Calculations Behind Lottery Valuations*

Hanyao Zhang†

May 23, 2025

Abstract

This paper introduces a novel experimental design tracking subjects' calculations when valuing lotteries with a calculator. Subjects predominantly employ simple functional forms, primarily expected values or linear functions of monetary outcomes. The calculations exhibit remarkable within-subject stability alongside substantial between-subject heterogeneity. Calculations strongly predict risk attitudes: subjects calculating expected values exhibit near risk-neutrality, while others display extreme unresponsiveness to probability changes and are consistent with Tversky and Kahneman's (1992) fourfold pattern. Notably, these calculations also predict subjects' valuations of deterministic mirrors (Oprea, 2024*b*), as well as the lottery valuations when not provided with a calculator. Leveraging the calculation data, I examine the mechanisms behind the observed fourfold pattern and unresponsiveness by: (1) distinguishing implementation costs from misconceptions, finding evidence suggesting each is linked to different subject types, and (2) evaluating probability weighting and cognitive imprecision models against observed calculations.

*I am indebted to Mark Dean for his invaluable guidance and unwavering support. I also benefited from comments by Hassan Afrouzi, Alessandra Casella, Navin Kartik, Judd Kessler, Zhi Hao Lim, Kirby Nielsen, Ryan Oprea, Mike Woodford, Yangfan Zhou, members of the Columbia Cognition and Decision Lab, seminar audience at Columbia, and SWEET (UPenn). Hao Lin and Chris Yuan provided excellent research assistance. I am grateful for the financial support from the Columbia Experimental Laboratory for Social Sciences. This research was approved by Columbia IRB. All mistakes are my own.

†Department of Economics, Columbia University, New York. Email: hanyao.zhang@columbia.edu

1 Introduction

A common finding in the literature of risk attitudes is that economic agents' valuations over risky lotteries are unresponsive: When probabilities over the potential monetary outcomes change, the resulting changes of lottery valuations are smaller in magnitude than the changes of lottery expected values. The pattern of unresponsiveness unifies the fourfold pattern documented in Tversky and Kahneman (1992) (Blavatskyy, 2007):[1] risk aversion for gains and risk seeking for losses of high probability; risk seeking for gains and risk aversion for losses of low probability. The unresponsiveness is also a leading example of the broader theme of behavioral attenuation that has been observed across multiple decision-making domains (Enke et al., 2024).

Understanding the cause of this unresponsiveness is crucial for both theory and applications. If this pattern reflects genuine risk attitudes (economic agents' welfare-relevant rank ordering of lotteries), they are welfare-relevant preferences that should guide both economic theory and policy design. However, the recent literature has highlighted the role of cognitive complexity (costs or difficulty to make choices according to the genuine risk attitudes) in explaining this unresponsiveness. Particularly, Oprea (2024*b*) shows that unresponsiveness also arises when agents value the *deterministic mirrors* of the lotteries. For each lottery, its deterministic mirror is presented in the same disaggregated form – a sequence of monetary amounts and their weights – but pays out the corresponding lottery's expected value with certainty. The replication of unresponsiveness in the valuations of deterministic mirrors demonstrates that the observed unresponsiveness, in the context of the experimental data, stems partially from the cognitive complexity involved in processing disaggregated outcomes, rather than solely reflecting genuine risk attitudes.

Oprea's (2024*b*) intriguing result naturally stimulates the question: What calculations underlie subjects' unresponsive valuations? Identifying these calculations is a crucial step towards better understandings of unresponsiveness, since the calculations help economists validate existing models of decision-making under risk, and inform new models to better capture the calculations behind lottery valuations and other risky choices.

[1] To see how unresponsiveness unifies the fourfold pattern, I refer the Reader to the discussion in Section 3.

Attempting to shed light on this question, I conduct an online experiment implementing a novel *calculator design* to elicit subjects' valuations of lotteries and their deterministic mirrors. The design and experimental instructions[2] are similar to Oprea's (2024*b*) except for one aspect: The design provides subjects with a calculator on the experimental interface when they value lotteries and mirrors, and tracks the calculations the subjects perform with the calculator. These calculations reveal the procedural and computational aspects of the decision-making process leading to the valuations.

The valuation data from the calculator design reproduces Oprea's (2024*b*) main finding: Unresponsiveness arises in the valuations of both lotteries and their deterministic mirrors. In addition to the main treatment implementing the calculator design (*Calc*), I also implement another within-subject treatment that drops the calculator from the experimental interface when the subjects value lotteries and mirrors (*NoCalc*). Comparing the Calc and NoCalc treatments, I show that subjects are more responsive with a calculator, but only by a small magnitude. This finding shows that the calculator design is suitable for studying unresponsive lottery valuations.

Next, I turn to the calculations performed by the subjects. I construct two features of the calculations as the main objects of study, both designed to capture the functional forms of the task primitives (the sequence of monetary amounts and their corresponding weights that define a lottery and its mirror) that the subjects calculate. The first feature, *procedure*, captures the functional forms mapping task primitives to valuations, while the second feature, *base term*, summarizes *all* functional forms of task primitives that the subjects calculate, not just the ones that can be directly linked to the valuation. Using these features, I conduct three main types of analysis: 1) A descriptive analysis of calculations performed by the subjects when valuing lotteries and mirrors; 2) A correlational analysis between subjects' calculations and the unresponsiveness in their valuations; and 3) A mechanism analysis distinguishing between competing mechanisms that might explain unresponsiveness in lottery valuations.

First, I answer the descriptive question – what do subjects calculate when they construct

[2]Addressing the concerns raised by Banki et al. (2025) and Wu (2025) over Oprea's (2024*b*), and by extension, my experimental instructions, I replicate all the analysis below restricting to those subjects who perfectly answer all comprehension questions. All results are robust to this sample restriction. See Appendix D for a more thorough discussion over the robustness of my results.

their lottery valuations? I find that in the vast majority of lottery tasks, subjects use a procedure that maps the task primitives into one of the following functional forms: the expected value (34.7%), a number not directly linked to any task primitives (31.9%), or a linear function of the potential monetary outcomes (20.6%). Moreover, the calculations are stable within-subject. That is to say, a fixed subject typically uses similar calculations to approach different tasks, even if across lottery and mirror tasks.

Second, I find the calculations to be strongly predictive of lottery valuations. Specifically, I use calculations performed by subjects when they value lotteries to categorize subjects into types, and investigate the responsiveness of the valuations of each type. This analysis leads to a striking dichotomy of subjects. Subjects who generally calculate the average payouts – the expected value (EV) type (38.1% of all subjects) – tend to submit very responsive valuations and appear close to risk neutral. To quantify this responsiveness, I regress the absolute valuations on the absolute expected values, separately using the lottery valuations for each subject, and use the regression coefficients of the expected values as a subject-level quantitative measure of the responsiveness. The average responsiveness of the EV-type subjects is 0.85. In other words, when the expected payout of the lottery increases by $1, on average the valuation of the lottery increases by $0.85. On the other hand, subjects who generally do not calculate the average payout – all the other non-EV types (61.9% of all subjects) – submit very unresponsive valuations. Their average responsiveness is merely 0.25. Putting this extreme unresponsiveness into perspective, These subjects' valuations change only minimally as probabilities change: their average abosolute valuation of lotteries G8 (8% chance of gaining $26) and L8 (8% chance of losing $26) is $11.4, while their average valuation of lotteries G92 (92% chance of gaining $26) and L92 (92% chance of losing $26) is only $16.4, despite the large difference in expected values. These subjects also appear well-described by the fourfold pattern. Perhaps most strikingly, most non-EV-type subjects exhibit close to zero or even negative responsiveness, strongly challenging the genuine risk attitudes interpretation.

Interestingly, the calculations made for lottery tasks have out-of-sample predictive power. Specifically, these calculations can also predict: 1) the *mirror* valuations of the same subject; and 2) the lottery valuations of the same subject *in the NoCalc treatment*. Related to the first, for the EV-type subjects who calculate the average payouts when valuing *lotteries*, their average responsiveness in *mirror* tasks is 0.82. In contrast, for the non-EV-type subjects who do not calculate the average payouts when valuing *lotteries*, their average responsiveness in *mirror* tasks is merely 0.24. Related to the second out-of-sample test, calculations made by a subject in the Calc treatment also predict the lottery valuations of the same subject in the NoCalc treatment. Especially, the EV-type subjects exhibit higher responsiveness than the non-EV-type subjects in the NoCalc treatment. This result is reassuring, as it suggests that the collected calculations succeed in capturing some aspects of the subjects' approaches in the more natural setting of the NoCalc treatment.

Further comparing the responsiveness in lottery tasks between the Calc and NoCalc treatments by subject types, I show that the slightly higher responsiveness in the Calc treatment is entirely due to the EV-type subjects. These subjects exhibit moderate responsiveness in the NoCalc treatment (responsiveness = 0.56), but high responsiveness in the Calc treatment (responsiveness = 0.85). Moreover, when given a chance to revise their choices when they differ across the two treatments (similar to Nielsen and Rehbeck (2022)), these EV-type subjects generally prefer their responsive valuations in the Calc treatment over their unresponsive valuations in the NoCalc treatment. These findings emphasize the role played by complexity in the experimentally measured risk attitudes – if giving some subjects a calculator changes them from having moderate responsiveness and appearing well-described by the fourfold pattern to being near completely responsive and appearing close to risk neutral, the apparent unresponsiveness and fourfold pattern must not reflect genuine risk attitudes. On the other hand, the non-EV-type subjects exhibit similar responsiveness in both Calc and NoCalc treatments, such that their responsiveness is not statistically distinguishable between treatments.

How do these findings inform our understanding of risk attitudes? The suite of evidence from this study echoes the finding of Oprea (2024*b*) that complexity plays a major role in the measured risk attitudes. First, not only are the *valuations* similar between lotteries and mirrors, but the *calculations* are also similar. Second, giving the EV-type subjects a calculator makes their lottery valuations almost risk neutral. Third and perhaps most importantly, the rather extreme unresponsiveness in the lottery valuations of the non-EV types casts skepticism over the possibility that their valuations, and the non-EV calculations behind these valuations, actually reflect their risk attitudes.[3]

Finally, I explore potential mechanisms behind unresponsive lottery valuations from two distinct perspectives. In the first perspective, to the extent that complexity can explain unresponsiveness, I examine whether this complexity stems from implementation complexity (costs of executing known optimal procedures despite awareness of them) or representation complexity (fundamental misconceptions about the valuation task). The calculations reveal patterns consistent with both mechanisms, but in different types of subjects: EV-type subjects are significantly more responsive when provided with a calculator – a pattern difficult to reconcile with genuine risk attitudes, as their choices appear to change merely with the availability of a calculator. Instead, this pattern is consistent with implmentation complexity. In contrast, a subset of subjects frequently use decreasing procedures (procedures that generate lower valuations when probabilities or monetary amounts increase), suggesting fundamental misconceptions about lottery valuation rather than expressions of genuine risk preferences.

In the second perspective, which shifts from broad categories of complexity to specific theoretical mechanisms, I take models of probability weighting and cognitive imprecision, and assess their predictions against actual calculation processes. More specifically, I recover the probability weighting function implied by *calculations* – contrasting with traditional approaches that infer probability weights from *valuations*. The recovered function shows an extremely flat slope (0.093), where subjects apply similar weights across widely different probabilities. I apply the same methodology to examine the theory of cognitive imprecision (Khaw, Li and Woodford, 2021). In both cases, the patterns observed in calculation data are difficult to reconcile with these theoretical accounts due to extreme degree of unresponsiveness, suggesting subjects may employ simpler heuristics (such as attribute substitution in Kahneman and Frederick (2002)) rather than engaging in probability weighting or cognitive imprecision as described by these models. This analysis comes with the important caveat that it requires

[3] Even if subjects use similar valuations and calculations for lotteries and mirrors due to their confusion of the two payment rules, as suggested by Banki et al. (2025) and Wu (2025), the *lottery* valuations of non-EV types should still reflect their genuine risk attitudes. Non-EV-types subjects' extremely unresponsive valuations cast skepticism on that claim.

treating these models as literal descriptions of decision-making procedures rather than "as-if" models.

The rest of this paper is organized as follows. Section 2 describes my experimental design. Section 3 describes the patterns of valuations. Section 4 outlines the methodology of analyzing the calculations, and provides descriptions of the calculations arising in my experimental data. Section 5 links the calculations to the valuations, and Section 6 discusses the implications of the calculations over the mechanisms behind unresponsive lottery valuations. Finally, Section 7 discusses how the current work relates to the literature.

2 Experimental Design

2.1 Lotteries and Their Deterministic Mirrors

In the experiment, I elicit the valuations (certainty equivalents) for a set of 8 distinct lotteries using the Becker-Degroot-Marschak (BDM) mechanism (Becker, Degroot and Marschak, 1964). I focus on simple, two-outcome lotteries where some amount $\$X$ is paid with probability p, and \$0 is paid with probability $1-p$. Two groups of lotteries are included. In the first group, "gain lotteries," $X = 26$ and the subject gains \$26 with probabilities $p \in \{0.08, 0.25, 0.75, 0.92\}$. These lotteries are referred to as G8, G25, G75, and G92, respectively. In "loss lotteries," $X = -26$ and the subject loses \$26 with probabilities $p \in \{0.08, 0.25, 0.75, 0.92\}$. These lotteries are referred to as L8, L25, L75, and L92, respectively. To measure valuation inconsistency and its relationship with valuation patterns, the lottery G25 is repeated, leading to a total of 9 lottery tasks.

The experimental instructions follow the BDM treatment in Oprea (2024*b*), depicting the gain (loss) lottery Gn (Ln) as 100 boxes, of which n contain \$26 ($\$-26$) and the rest contain \$0. To determine the payment from the lottery, one of these boxes will be randomly selected, and the amount of money in the selected box will be paid.

Again following Oprea (2024*b*), I elicit the valuations of the *deterministic mirror* of each lottery, with the mirror of lottery G25 again repeated twice. A deterministic mirror is presented in a similar format as its corresponding lottery, but features a modified payoff rule that eliminates risk and pays the expected value of its corresponding lottery with certainty. Specifically, a mirror is also depicted as 100 boxes, each containing some amount of money. However, instead of paying out a randomly selected box like a lottery does, a mirror pays the average amount of money across the 100 boxes. I use tuples such as (G8, lottery) and (L25, mirror) to refer to individual valuation tasks, and use *task type* to refer to the two different payoff rules: Lottery and mirror.

2.2 Experimental Treatments

The experiment includes two within-subject treatments: *NoCalc* and *Calc*.

NoCalc Treatment In the NoCalc treatment, the subject values the 9 lotteries and mirrors by typing their valuations into a text box. The subject is given \$30 as their initial money for them to bid under the BDM mechanism, and their gains and losses from the lotteries and mirrors are calculated on top of the initial money. The valuations are restricted to be between \$0 and \$26 for tasks involving gains, and between \$−26 and \$0 for tasks involving losses. The NoCalc treatment replicates Oprea (2024*b*) with minimal changes of parameters and offers a benchmark with which the valuations in the main Calc treatment can be compared.

Calc Treatment In the Calc treatment, the subject values the same 9 lotteries and mirrors. The key innovation is the inclusion of a calculator in the experimental interface, whose input I can track and record (the calculator design). The calculator can perform basic arithmetic operations (addition, subtraction, multiplication, division). Numbers and operations can be typed into the calculator by either clicking the buttons on the graphical interface or using a keyboard. Moreover, the calculator can store multiple calculations. All previous expressions calculated are displayed in the calculator as a table, in the order of being performed. Each line in the table consists of a *Calculation* column, where the expression calculated is displayed, and a *Result* column, where the calculated result appears. The calculator refreshes after each task, clearing all previous expressions and results. A screenshot of the experimental interface with example calculations performed can be seen in Figure 1.

As in the NoCalc treatment, a text box is provided asking for the valuation of the subject

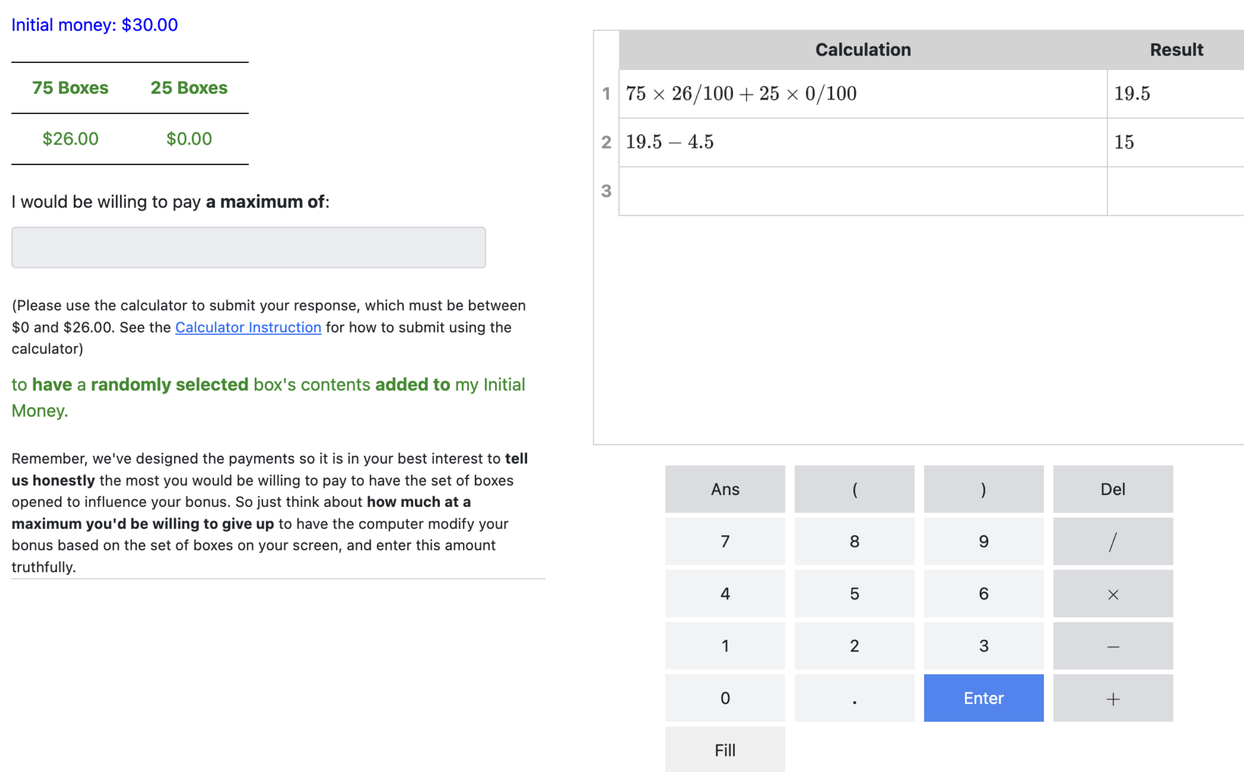


Figure 1: The experimental interface in the Calc treatment with example calculations performed

(seen on the left of Figure 1). However, the text box is grayed out, and the subject is not able to directly type numbers into the text box. Instead, to submit a valuation, the subject needs to make it appear in the Result column of the last line of the calculator. Then, the subject can click the *Fill* button (shown in the bottom-left of the calculator interface in Figure 1) to fill the number into the grayed out text box. This submission process is designed to balance two goals. First, it gently encourages subjects to use the calculator. Second, it minimizes potential distortions of behaviors. Particularly, the design accommodates subjects who wish to submit valuations without performing calculations in the calculator. Such a subject can simply type their intended valuation as a single number in the calculator's first line, and then "calculate" this number. The number will appear in the Result column, allowing them to use the *Fill* button to submit their valuation.

The subject does not receive any specific instructions as to how the calculator may help them in the task, and they are free to use the calculator to perform whatever calculations they deem useful. The payment of the subject does not depend on what the subject types into the calculator, and only depends on the valuations that they submit.

Timeline The NoCalc treatment consists of two blocks – one containing all lottery tasks and the other containing all mirror tasks. The order of blocks and the order of tasks within each block are randomized at the subject level. The subject is not informed about the second block while completing the first block, but receive instructions about the new task type before beginning the second block.

After finishing the NoCalc treatment, the subject enters the Calc treatment. Within the Calc treatment, there are again two blocks for lottery tasks and mirror tasks, respectively. The order of lottery and mirror blocks in the Calc treatment is the same as that in the NoCalc treatment. Figure 2 shows the diagram for the main experimental timeline.

Before each of the four blocks, the subject is required to answer three comprehension questions taken from Oprea (2024*b*). These comprehension questions serve the purpose of training the subjects over the payoff rules in lottery tasks and mirror tasks, and also as a reminder that the payoff rule has changed from the one used previously. The subject has unlimited opportunities to answer the questions.

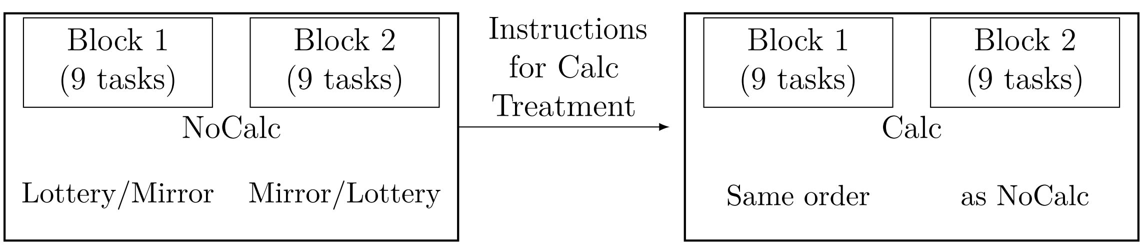


Figure 2: Main Experimental Timeline

After the Calc treatment, following Nielsen and Rehbeck (2022), the subject is asked to reconcile their potentially different valuations for the same task between NoCalc and Calc treatments. Finally, the subject responds to a few additional questions, including an incentivized choice among four deterministic mirrors. Analysis for the reconciliation stage and the additional questions is reported in Appendix E.

2.3 Implementation Details

The experiment was conducted on Prolific in January 2025. A total of 202 subjects completed the experiment. The experiment was programmed using OTree (Chen, Schonger and Wickens, 2016). Each subject was paid a participation fee of $7 for completing the experiment. With a 20% chance, a subject was also paid the outcome of a randomly chosen task. The median subject spent around 50 minutes on the experiment, and the average total earning from the experiment was $13.19. The experimental instructions can be found in Appendix G.

3 Valuations of Lotteries and Mirrors

This section focuses on the valuations of lotteries and mirrors submitted by the subjects. Analysis of the calculations is left to the following sections.

Figure 3's four panels show the average absolute valuations for all lotteries and mirrors in both NoCalc and Calc treatments, pooling across all subjects. First, the lottery valuations exhibit substantial unresponsiveness – when the expected values of lotteries change, the lottery valuations change by a smaller magnitude. This pattern of unresponsiveness unify the classic fourfold pattern documented in Tversky and Kahneman (1992). To see this, note that unresponsiveness implies a pull-to-the-center effect in the valuations. When the probability of the non-zero outcome is small, the absolute valuations are greater than the absolute expected values for both gains and loss lotteries, indicating patterns conventionally interpreted as risk-loving preferences for small probability gains, and risk-averse preferences for small probability loss.[4] In contrast, when the probability of the non-zero outcome is large, the relationship between the absolute valuations and the expected values reverse. As a result, the average subject appears to have risk-averse preferences for large probability gains, and risk-loving preferences for large probability loss.

Second, replicating Oprea (2024*b*), mirrors exhibit a similar pattern of unresponsiveness as lotteries.[5] To formally test whether lottery and mirror valuations differ, I conduct Kolmogorov-

[4]For loss lotteries, since both valuations and expected values are negative, absolute valuations being greater than absolute expected values means valuations are smaller (more negative) than expected values, a pattern conventionally interpreted as risk-averse preferences.

[5]The magnitude of unresponsiveness in valuations observed in the comparable NoCalc treatment of this

Smirnov tests comparing the distributions of valuations for each lottery with its corresponding mirror. This yields eight p-values per treatment (one for each lottery-mirror pair). In the NoCalc treatment, none of these p-values is below 0.1. In the Calc treatment, only one p-value falls below 0.1. After applying standard multiple hypothesis testing corrections to control the familywise error rate (such as the Holm-Bonferroni method), I fail to reject the null hypothesis of equal distributions for any lottery-mirror pair in either treatment.

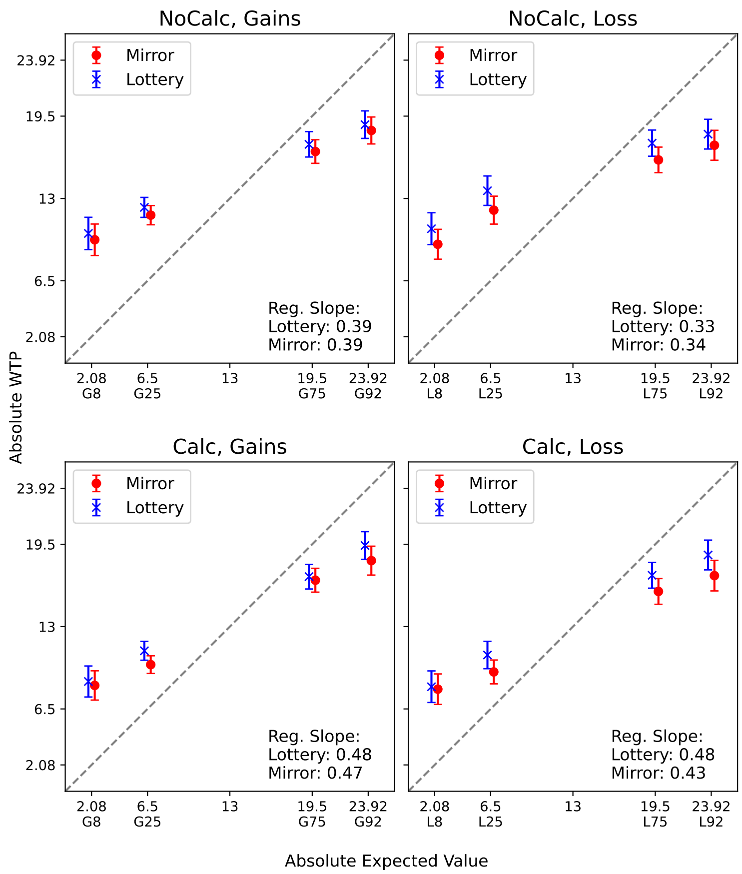


Figure 3: Average valuation of each lottery and their deterministic mirror. 95% confidence interval are shown with the bars, and the 45-degree line is shown as a dashed line.

Third, contrasting the findings from the re-analysis of Oprea's (2024*b*) data by Banki et al. (2025), this pattern of similar lottery and mirror valuations holds not only when aggregating across all subjects, but also when looking at the subset of 49 subjects who correctly answer all four sets of comprehension questions at their first trial. When restricting the data to these subjects and conducting the same set of 16 K-S tests above, I again cannot reject any

study is quantitatively similar to that reported in Oprea (2024*b*).

of these null hypotheses after applying standard multiple hypothesis testing corrections.

Fourth, the absolute valuations are similar for gain and loss lotteries with the same probability of non-zero outcome, for example G8 and L8.[6] To simplify the analysis, from now on, I treat pairs of gains and loss lotteries with the same probability of non-zero outcome (such as G8 and L8) as the same lottery. That is to say, for a loss lottery, I take absolute values for any characteristics related to it, such as valuation, expected value, and potential monetary outcomes. In the following text, I will simply use the terminology *valuation* to refer to the absolute valuation, and similarly for other characteristics. Similarly, I treat analogous pairs of gains and loss mirrors as the same mirror. Analyzing gains and loss tasks separately does not meaningfully alter any of the results in this paper.

Finally, I examine whether providing access to a calculator affects subjects' lottery and mirror valuations. Figure 3 shows that providing a calculator to subjects does not eliminate the unresponsiveness in their lottery and mirror valuations. However, it is also clear from the slopes shown in the graph that valuations in Calc are somewhat more responsive than those in NoCalc. This difference is statistically significant. I will return to the comparison between NoCalc and Calc in Section 5.3 with a more granular analysis focused on subsets of subjects.

4 Descriptions of Calculations

This section first outlines the methodology for analyzing the calculation data, and then provides descriptive statistics of the data.

4.1 The Data: Calculator Inputs

Since this paper analyzes a novel type of data – the sequence of calculations performed by subjects using the calculator – I first provide a brief overview of the structure of the data.

For each task completed by each subject in the Calc treatment (referred to as a *round*), the computer collects two main pieces of data. First, it collects the subject's *valuation* of this lottery or mirror. Second, it also collects the *calculator input* of the subject. The

[6]The same pattern repeatedly appears in experiments measuring lottery valuations. See, for example, Tversky and Kahneman (1992, Table 3, page 307), l'Haridon and Vieider (2019, Figure 2, page 196).

calculator input is a sequence of numerical expressions typed into the calculator by the subject and evaluated by the calculator. The order of numerical expressions in the calculator input reflect the order by which the subject performs them. The calculator input data also contains a sequence of *results*, each derived from evaluating the corresponding expression in the calculator input.

Table 1 visualizes a hypothetical observation of calculator input, where a subject faces the task (G75, Lottery). The numerical expressions in the calculator input and the numbers in the results are ordered by their *line*, which reflects the order by which the subject calculates them. This calculator input reveals that, when facing the task, the subject first typed $75 \times 26/100 + 25 \times 0/100$ into the calculator and evaluated it, and then did the same for $19.5 - 4.5$. Since the experimental design requires the valuation to appear in the Result column of the last line of the calculator before it is submitted (See descriptions for the Calc treatment in Section 2.2), the result in the last line, 15, is also the valuation submitted by the subject.

Line	Numerical Expression	Result
1	$75 \times 26/100 + 25 \times 0/100$	19.5
2	$19.5 - 4.5$	15

Table 1: An example calculator input by a hypothetical subject facing the task (G75, Lottery).

Analyzing the calculator input data raises a few challenges. First, calculator inputs are high-dimensional objects and cannot be directly used in a quantitative analysis. Second, the calculator inputs only represents the "lower bound" of all calculations performed by the subjects. In other words, calculations performed in the mind of a subject will not show in the calculator input data. Finally, to rigorously define the objects and describe the algorithms involved in the analysis, a formal structure of mathematical expressions is needed. An introduction to the formal structure and rigorous descriptions of the algorithms used in this section are provided in Appendix F. In what follows, I rely on examples to illustrate the definitions and algorithms.

4.2 Recovering Symbolic Expressions

While subjects can only calculate numerical expressions in the calculator, their corresponding *symbolic expressions* (in terms of the task primitives) can be recovered to shed light on the calculation strategies by which subjects reach their valuations. Four primitives describe each task: b_1 denotes the number of boxes with non-zero amount of money, m_1 denotes the (absolute) amount in each of these boxes, b_2 denotes the number of boxes with zero amount, and m_2 denotes the amount in each of these boxes. For example, when facing the task (G75, Lottery), where these primitives take values $b_1 = 75$, $b_2 = 25$, $m_1 = 26$, $m_2 = 0$,[7] a subject who calculates 75×26 is actually computing $b_1 \times m_1$. This symbolic representation reveals how subjects utilize task primitives to construct their valuations and facilitates comparisons across different tasks. For example, if the same subject calculates 25×26 when facing the task (G25, Lottery), recovering the same symbolic expression $b_1 \times m_1$ identifies the fact that, although the numerical expressions are different, the underlying calculations performed by the subject are the same as the previous example.

To recover the symbolic expressions, I use a simple match-and-replace algorithm. There are three key features of this algorithm. First, the algorithm processes the calculator input sequentially from the first line to the last line. Second, the algorithm matches the numbers in the numerical expressions against the task primitives. A number that matches a task primitive is replaced with the corresponding symbol. Third, taking into consideration the sequential nature of how subjects perform calculations, starting from the second line, the algorithm also matches the numbers against the set of previous results. A number that matches a previous line result is replaced with the recovered symbolic expression of that line. Any number that does not match any primitive or previous result remains as the same number.

Here, the algorithm recovering is illustrated using the example in Table 1. As the first step, the algorithm matches the numbers in the numerical expression in line 1 against the set of primitives for the task (G75, Lottery): $\{(b_1, 75), (b_2, 25), (m_1, 26), (m_2, 0)\}$.[8] Replacing the

[7] While the numbers corresponding to m_1 (26) and m_2 (0) do not change across rounds, the symbols are still useful since they make it much easier to see the symbolic calculation. In other words, they make it easy to identify, for example, a subject is calculating the expected value.

[8] When implementing the algorithm, the set of primitives that the algorithm matches against are expanded

matched numbers with their corresponding symbols leads to the symbolic expression of line 1, $b_1 \times m_1/100 + b_2 \times m_2/100$, where the number 100 doesn't find a match in the primitives and is left intact. Next, the algorithm moves to line 2. The number 19.5 does not match any task primitive, but it matches the result of line 1. Thus, the algorithm replaces the number 19.5 with the recovered line 1 symbolic expression, $b_1 \times m_1/100 + b_2 \times m_2/100$. The number 4.5 does not find a match and is kept the same. In this way, the algorithm recovers the line 2 symbolic expression, $b_1 \times m_1/100 + b_2 \times m_2/100 - 4.5$. The results are shown in Table 2.

Line	Numerical Expression	Result	Symbolic Expression
1	$75 \times 26/100 + 25 \times 0/100$	19.5	$b_1 \times m_1/100 + b_2 \times m_2/100$
2	$19.5 - 4.5$	15	$b_1 \times m_1/100 + b_2 \times m_2/100 - 4.5$

Table 2: Symbolic expressions of the example in Table 1, where a hypothetical subject faces the task (G75, Lottery).

The recovered symbolic expressions are the "lower bound" of all calculations performed by the subjects when valuing the lotteries and mirrors. They only capture the explicit calculations performed in the calculator, but not any implicit calculations performed consciously or subconsciously in the subjects' minds. This point can be seen clearly by comparing the two examples Table 2 and Table 3. The subject in Table 3 performs the same expected value calculation as Table 2. However, after computing the expected value, instead of explicitly performing the subtraction of 4.5 from the expected value as in Table 2, they perform this fairly simple calculation in their head, and types 15 directly into the second line and submit this as their valuation. Now, in Table 3, the number 15 in line 2 cannot be matched to any task primitives or previous results. As a result, the symbolic expression is simply the constant function 15, and the connection between the number 15 and the task primitives cannot be seen from this recovered symbolic expression. This is an important limitation of the calculator input data, and the two different features of the calculator input data in Section 4.3 reflect trade-offs involved in dealing with this limitation.

to include common calculation shortcuts that subjects may use. For example, in G75, the primitive set also includes $(b_1/100, 0.75)$ to capture the scenario where a subject divides the number of boxes by 100 implicitly in their mind before they use the result of this mental calculation directly in the calculator. For more detailed information on the set of primitives used, see Appendix F.

Line	Numerical Expression	Result	Symbolic Expression
1	$75 \times 26/100 + 25 \times 26/100$	19.5	$b_1 \times m_1/100 + b_2 \times m_2/100$
2	15	15	15

Table 3: An example calculator input by a hypothetical subject facing the task ($G75$, *Lottery*).

4.3 Procedures and Base Terms

Because the calculator inputs are high-dimensional objects, I develop two complementary features to facilitate the analysis. Both features rely on the recovered symbolic expressions, but they differ in terms of their emphasis. First, I construct the *procedure* to capture how the subject *maps task primitives to their final valuation*. Second, I construct the *base term set* to summarize *all* functional forms of task primitives a subject calculates in a round.

Procedures For a round, its procedure is defined as the recovered symbolic expression of the last line of the calculator input.[9] Since the experimental design requires the valuation to appear in the Result column of the last line of the calculator before it is submitted, the procedure is the function mapping task primitives to the final valuation. For example, in the calculator input shown in Table 2, its procedure is $b_1 \times m_1/100 + b_2 \times m_2/100 - 4.5$.

Ideally, the definition of procedures attempts to capture the function that the subject uses to map the primitives of the task to their valuations. However, the example illustrated in Table 3 has shown a potential pitfall of this attempt. In Table 3, the procedure itself, being the constant function 15, does not reflect the subject's use of task primitives in their calculations. The procedure is silent on the fact that the subject also calculates the expected value of the lottery, and may have used the calculated expected value in a way that is not captured by the calculator input data to construct their final valuation of \$15. To address this problem, I introduce base terms to complement procedures in the analysis of calculator inputs.

[9] More precisely, the procedure is the *function of task primitives* represented by the recovered symbolic expression of the last line of the calculator input. This lengthy definition emphasizes the fact that a procedure is a function, not an expression. In other words, two expressions with different syntaxes but representing the same function, for example, $2 \times b_1 \times m_1/200$ and $(b_1 \times m_1)/100$, should be viewed as the same procedure.

Base Terms and Base Term Sets I decompose a calculator input into its base terms: All terms that appear in the symbolic expressions of the calculator input, with their numerical factors dropped. The process of identifying base terms involves three steps:

1. First, I break down each symbolic expression in the calculator input into terms.[10] For example, in the symbolic expression $b_1 \times m_1/100 + b_2 \times m_2/100$, there are two terms: $b_1 \times m_1/100$ and $b_2 \times m_2/100$.[11]

2. Next, I generate a base term from each term by dropping all numerical factors. For example, dropping the numerical factor $1/100$ from term $b_1 \times m_1/100$ generates its base term $b_1 \times m_1$. If a term contains only numerical factors and no symbolic factors (for example, the terms 4 and 4×2), its base term is defined as C.

3. Finally, I define the *base term set* of a calculator input as the collection of all base terms that appear in any of its symbolic expressions.

This definition is illustrated again using the example in Table 1. In the symbolic expression of line 1 ($b_1 \times m_1/100 + b_2 \times m_2/100$), two base terms appear: $b_1 \times m_1$ and $b_2 \times m_2$, whereas in the symbolic expression of line 2 ($b_1 \times m_1/100 + b_2 \times m_2/100 - 4.5$), a total of three base terms appear: C in addition to $b_1 \times m_1$ and $b_2 \times m_2$. Therefore, the base term set of this example calculator input is $\{b_1 \times m_1, b_2 \times m_2, C\}$.

The base terms provide a summary of the functional forms of *all calculations*, while the procedures capture the exact functional form of the *submitted valuation*. These two features both reduce the dimension of the calculator input, and in the meantime complement each other. First, since all number factors are dropped when constructing the base terms,

[10] The concept of terms, and by extension base terms, runs into problems with syntactically different but mathematically equivalent expressions – for example, the mathematically equivalent expressions $b_1 \times m_1/100 + b_2 \times m_2/100$ and $(b_1 \times m_1 + b_2 \times m_2)/100$ lead to different base terms. To solve this problem, I first expand all the products in all expressions by applying the distributive law of multiplication ($a \times (b + c) = a \times b + a \times c$), wherever applicable. This way, I transform the original symbolic expression into its distributed form expression. The base term set of a calculator input is defined as the collection of all base terms that appear in any of its distributed form expressions. Using distributed form expressions solves the aforementioned indeterminacy problem and generates the same set of base terms for $b_1 \times m_1/100 + b_2 \times m_2/100$ and $(b_1 \times m_1 + b_2 \times m_2)/100$.

[11] Division operations in expressions are represented as multiplication operations. For example, the expression $b_1 \times m_1/100$ is transformed to $b_1 \times m_1 \times \mathrm{Inv}(100)$, where $\mathrm{Inv}(100)$ denotes the inverse of 100, a number. For expositional simplicity, I still keep the division sign in expressions. Similarly, subtraction operations in expressions are transformed to addition operations.

procedures complement base terms by preserving the information in the unmatched numbers. Second, since the procedures lose any information over the calculations that cannot be directly connected to the valuation, base terms complement procedures by preserving the information in these calculations. For instance, though the examples in Table 2 and Table 3 generate different procedures, they generate the same base term set $\{b_1 \times m_1, b_2 \times m_2, C\}$, which emphasizes the similarity of these two calculator inputs.

Procedure Groups and Base Term Groups For parsimony and interpretability, I categorize base terms into five groups. Beyond its intuitive appeal, this categorization is also supported by fitting a unsupervised machine learning topic model on the calculator inputs.[12] The five groups are:

1. **Expected value**: $b_1 \times m_1$, $b_2 \times m_2$;
2. **Number**: C;
3. **Linear box**: b_1, b_2;
4. **Linear money**: m_1, m_2;
5. **Non-linear**: All other (non-linear) terms, for example $b_2 \times m_1$ and b_2/b_1.

Parallel to the base term groups, I also classify all procedures into five groups.

1. **Expected value**: Procedure is the expected value function $b_1 \times m_1/100 + b_2 \times m_2/100$ or $b_1 \times m_1/100$.
2. **Number**: The procedure is a constant function that does not depend on the task primitives. It assigns a fixed numerical value irrespective of the input.

   Example: 5.

[12] I use Latent Dirichlet Allocation (LDA, Blei, Ng and Jordan, 2003) to find latent *topics* from calculator inputs, and group base terms by the topic that they are strongly associated with. The five groups of base terms listed in the main text are each associated with an individual topic. This exercise draws an analogy between calculator inputs and text documents in nature language. LDA is a popular unsupervised technique in natural language processing to find latent semantic topics from text documents. Details of this can be found in Appendix B.

3. **Linear money**: The procedure is a linear function of the two potential monetary outcomes, expressed as $\gamma_0 + \gamma_1 \times m_1 + \gamma_2 \times m_2$, where $\gamma_0, \gamma_1, \gamma_2$ are constants.
   Example: $0.7 \times m_1 + m_2/10$.

4. **Linear box**: The procedure is a linear function of the two numbers of boxes, written as $\theta_0 + \theta_1 \times b_1 + \theta_2 \times b_2$, where $\theta_0, \theta_1, \theta_2$ are constants.
   Example: $0.1 \times b_1 + b_2/5$.

5. **Non-linear**: Any procedure that does not match the descriptions of any group above.
   Example: $b_1 \times m_2/100$.

4.4 Descriptions of Calculator Inputs

Now, I provide descriptions of the calculator inputs appearing in the experiment. First, in 75% of all rounds, some explicit calculations are performed, while in the remaining 25% of all rounds, no explicit calculation is performed at all, and the subjects simply type a number into the calculator and submit this number.

The next question I aim to answer is: What calculations do subjects perform when they value lotteries and mirrors? Particularly, besides the calculations of the expected value, does there exist any other functional form of task primitives that is repeatedly calculated by different subjects? To answer this question, In Panel A of Table 4, I list all non-number procedures that appear in more than 0.5% of all rounds, in addition to their shares in the two task types separately. Except for the expected value procedures and a few linear money procedures, other non-number procedures appear in only a tiny share of rounds.

Looking at the base terms paints the same picture. Across the 3636 Calc rounds, only 44 distinct base terms appear, and only 8 of these appear in more than 1% of rounds (listed in Panel A of Table 5). These most frequently used base terms often have interpretable functional forms: the components of expected value calculations ($b_1 \times m_1$ and $b_2 \times m_2$), the number term C,[13] and the linear terms m_1, m_2, b_1, b_2. All unlisted base terms together appear in only 5.9% (6.7%) of lottery (mirror) rounds. In other words, even when given the flexibility

[13]Most of these number terms consist of only one unmatched number, such as 4, as opposed to a few unmatched numbers multiplied together, such as 4×2. More specifically, if I only look at the frequency of terms with only one number, they appear in 28.6% of lottery rounds and 27.9% of mirror rounds.

Panel A: Procedures	lottery	mirror	Panel B: Procedure Groups	lottery	mirror
$b_1 \times m_1/100$	29.0%	32.9%	expected value	34.7%	37.0%
m_1	11.3%	6.6%	number	31.9%	34.8%
$b_1 \times m_1/100 + b_2 \times m_2/100$	5.7%	4.1%	linear money	20.6%	14.8%
$m_1/2$	0.8%	2.8%	nonlinear	9.0%	9.4%
m_2	2.1%	0.9%	linear box	3.7%	4.1%
$b_2 \times m_1/100$	1.5%	1.0%			
$m_1/4$	1.0%	0.4%			
$b_1 \times m_1/200$	0.4%	1.0%			
$m_1 + m_2$	0.8%	0.6%			
$3 \times b_1/10$	0.7%	0.6%			
$100/b_1$	0.6%	0.7%			

Table 4: Frequencies of procedures and procedure groups. Panel A: Shares of all non-number procedures that appear in more than 0.5% of all rounds (lottery and mirror). Panel B: Shares of all five procedure groups (see the text for the definition).

to construct any calculation, subjects overwhelmingly restrict themselves to expected value or linear functions of task primitives.

Moreover, across lottery and mirror tasks, the shares of procedures and base terms are similar. The finding micro-founds the similar lottery and mirror valuations documented in Oprea (2024*b*) and in Section 3 of this paper.

Finally, and perhaps a bit surprisingly, even if expected value is the most commonly used base term group, it is only used in 43.7% of all mirror rounds, and this number is only slightly higher than its counterpart for lottery rounds. In other words, in mirror rounds, it is not the case that most subjects calculate the expected value but do not submit it as their valuation. Rather, in most of these rounds, the subjects do not calculate the expected values at all. This finding signals widespread non-optimizing behaviors.

Result 1. *In most rounds, the subjects do perform calculations when provided with a calculator, and they mostly calculate the expected value or linear functions of the task primitives when facing the valuation tasks. Aggregating across all subjects, calculations are similar between lottery and mirror tasks.*

Panel A: Base Terms	lottery	mirror	Panel B: Base Term Groups	lottery	mirror
$b_1 \times m_1$	40.6%	43.7%	expected value	40.7%	43.7%
C	38.0%	36.9%	number	38.0%	36.9%
m_1	20.1%	18.3%	linear money	23.4%	20.5%
$b_2 \times m_2$	7.0%	6.4%	linear box	7.8%	7.3%
b_1	6.5%	5.8%	nonlinear	7.1%	7.0%
m_2	4.9%	3.6%			
b_2	4.1%	3.4%			
$b_2 \times m_1$	1.9%	1.4%			
All others	5.8%	6.4%			

Table 5: Fractions of rounds where each base term (group) appear. Panel A: The fractions of all base terms that appear in more than 1% of all rounds (lottery and mirror). Panel B: The fractions of all five base term groups (see the text for the definition).
Note: In Panel A, $b_2 \times m_2$ is used much less often than $b_1 \times m_1$ because m_2 always corresponds a numerical value of 0, and subjects who realize this could skip using the base term $b_2 \times m_2$ without affecting their calculation results.

Examples of calculator inputs appearing in the data can be found in Appendix C.

4.5 Within-Subject Stability of Calculator Inputs

Next, I ask the following question: Does the same subject use stable calculations across tasks? There are two facets of this stability. First, I examine *within-task-type* stability of procedures: Does the same subject use the same procedure across tasks within the same task type? For each combination of subject and task type, I identify the *modal procedure*: The most frequently occurring procedure within that subject's calculator inputs for all rounds of that task type. I then compute what fraction of that subject's rounds use their modal procedure. Taking the median of these fractions across subjects shows that the median subject uses their modal procedure in 5 out of 9 rounds for both lottery and mirror tasks.

Using exactly the same procedure for multiple tasks requires that every step in the valuation process is explicitly performed in the calculator. If a subject performs part of their valuation process implicitly in their mind, their procedures as defined here will differ across rounds, but the actual valuation processes may still be similar. This argument manifests the values of conducting an additional analysis using the coarser notion of procedure groups, as opposed to the original procedures. Using the same procedure group in two rounds indicates generally similar approach to constructing the valuations, but allows the possibility of implicit calculations. An analogous analysis as what has been done above for procedures reveals that the median subject uses their modal procedure group in 8 out of 9 rounds for both lottery and mirror tasks. Another analogous analysis for base term sets reveals an equally strong pattern – for both lottery and mirror tasks, the median subject generates identical base term sets in 8 out of 9 rounds. That is, their calculator inputs in these 8 rounds contain precisely the same collection of base terms, no more and no less. These analyses demonstrate that the majority of subjects maintain a fairly stable procedure group and base term set for a given task type.

Second, I examine *between-task-type* stability: Does the same subject use the same procedure when valuing *a lottery and its corresponding mirror*? I refer to all lottery-mirror round pairs where the same subject faces a lottery and its matched mirror as *within-subject pairs*. I find that 44.9% of within-subject paired rounds have identical procedures, while as a benchmark, the analogous figure for all lottery-mirror round pairs (including pairs across different subjects) is only 11.2%. Looking at the coarser notion of procedure groups instead of the procedures, among within-subject pairs, 68.9% have identical procedure groups (benchmark using all lottery-mirror pairs: 28.0%). This figure is significantly higher than the one using procedures. Perhaps a bit unexpectedly, there is only a slight tendency for subjects to switch to EV-group procedures when facing mirrors. In 8.0% of all within-subject pairs, the subjects use an EV-group procedure for the mirror task but switch to a non-EV-group procedure for the lottery task, while in 5.8% of all these pairs, the subjects use an EV-group procedure for the lottery task but switch to a non-EV-group procedure for the mirror task. In a parallel analysis for base term sets, 59.6% of all within-subject pairs have identical base term sets (benchmark using all lottery-mirror pairs: 18.9%). These results demonstrate that subjects typically use similar approaches to valuing a lottery and its corresponding mirror, but the specific calculations sometimes differ.

Result 2. *The procedures and base term sets are generally stable within-subject either within lottery tasks, within mirror tasks, or between lottery tasks and mirror tasks. In short, a fixed* *subject generally uses similar calculations to approach different tasks.*

5 Linking Calculator Inputs to Valuations

The next question I study is the connection between the calculator inputs and valuations. Section 5.1 looks at the round-level correlation between calculator inputs and valuations. I categorize rounds by the base terms in their calculator inputs, and then examine the distributions of valuations conditional on employing different base terms. Next, in Section 5.2, I look at the subject level and aggregate across rounds. I assign types to subjects based on their calculator inputs in lottery tasks, and establish the ties between the types of subjects and their responsiveness in lottery valuations.

5.1 Round-Level Calculator Inputs and Valuations

Does the calculator input in a round predict the valuation in the same round? Figure 4 shows the cumulative distribution functions (CDF) of the valuations in lottery tasks in the Calc treatment, where each panel plots the CDF conditional on employing a group of base terms.[14]

Figure 4 offers a few salient observations. First, rounds where the subjects employ an EV term generate responsive valuations that discriminate lotteries with different probabilities of winning or losing – assigning higher absolute valuations for lotteries with higher probabilities of winning/losing. In contrast, conditional on employing a non-EV term, the valuations are unresponsive and do not discriminate across lotteries – assigning similar valuations for lotteries with different probabilities of winning. To make a more quantitative comparison, I calculate the Kolmogorov-Smirnov (K-S) distance between the distributions of the valuations of G92/L92 and those of G8/L8.[15] This K-S distance between the valuations of G92/L92 and G8/L8 is 0.94 conditional on employing an EV term, but is only 0.29, 0.26, and

[14]I focus on the distribution of lottery valuations conditional on base terms, as opposed to procedures, because base terms offer a more comprehensive picture of the calculator inputs.

[15]For two empirical distributions with CDFs F_1 and F_2, the K-S distance is defined as $D_{KS}(F_1(x), F_2(x)) = \sup_x |F_1(x) - F_2(x)|$, the maximum vertical distance between two CDFs. A close-to-one K-S distance indicates that $F_1(x)$ and $F_2(x)$ are dissimilar, in the sense that there exists a point x^* such that one of the distributions has nearly all of its mass above it, while the other has nearly all of its mass below it. A close-to-zero K-S distance indicates the opposite.

0.19 conditional on employing a number term, a linear money term, or a non-linear term respectively. Conditional on employing a linear box term, the K-S distance is 0.53, higher than other non-EV terms, though linear box terms only appear in 8% of all lottery rounds.

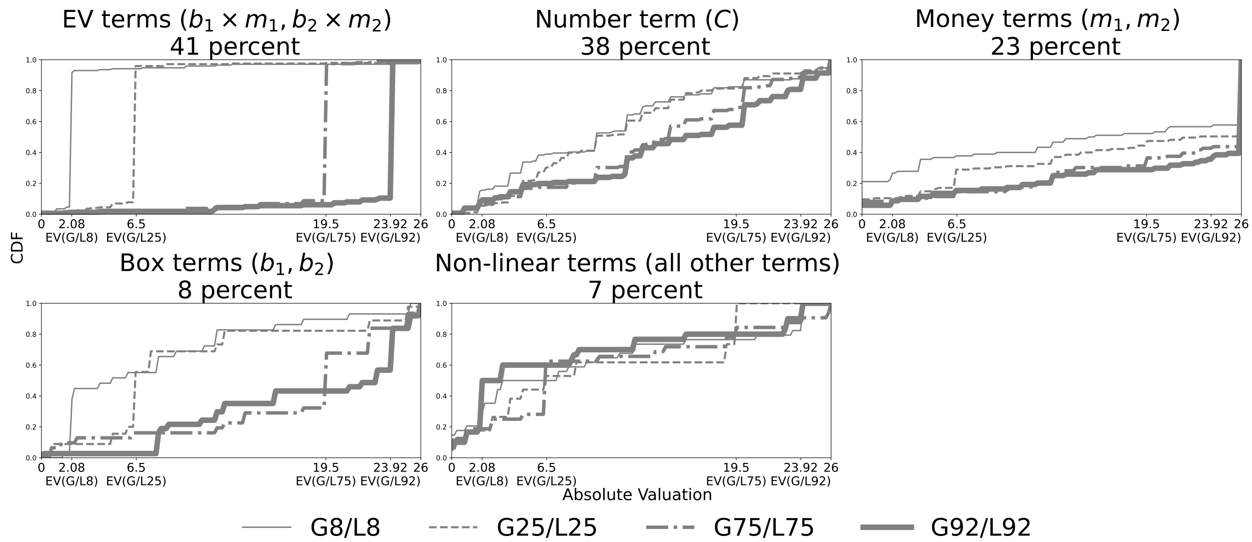


Figure 4: Distributions of lottery valuations in the Calc treatment, conditional on employing each group of base terms.

Second, the dispersion of valuations vary a lot conditional on employing different groups of base terms. Specifically, when an EV term is employed, the valuations tend to be highly concentrated around the expected value – around 88% of these rounds produce valuations exactly the same as the expected values. In contrast, when a non-EV term is employed, the distributions are much more dispersed. For a quantitative comparison, I focus on the interquartile range, i.e., the difference between the 25th and the 75th quantiles in the valuations, conditional on employing a group of base term and for a fixed lottery task. Conditional on employing an EV term, the interquartile range is exactly 0 for all four lotteries. However, for all other combinations of a non-EV term group and a particular lottery task, the interquartile range is much higher, between 6 and 23.92.

The analysis reveals that when subjects employ EV terms in a round, their resulting valuations tend to be risk neutral. In principle, even when EV terms are employed, valuations could still deviate from expected values through two mechanisms. First, since numerical factors are dropped when constructing base terms, a subject who calculates, for example, $0.9 \times b_1 \times m_1/100$ would produce a valuation that differs from the expected value. Second, since Figure 4 shows data conditional only on the presence of EV terms, subjects may simultaneously employ additional non-EV terms in their calculations. For instance, a subject might calculate $b_1 \times m_1/100 - 1$, and this round would still be included in the EV terms panel of Figure 4 despite the subtraction of a constant. Nevertheless, our data show that when EV terms are employed, valuations remain predominantly clustered around the expected value.[16]

Finally, conditional on each base term group, there are some idiosyncratic patterns. For example, while conditional on employing a number term the valuations are quite evenly distributed between \$0 and \$26, when instead conditional on employing a money term, the valuations have large point masses at \$26, even for the tasks G8/L8. Most strikingly, conditional on employing a non-linear term, the valuations of G8/L8 are systematically higher than those of G92/L92. This is the opposite of what one should expect, and is inconsistent with most mechanisms (e.g. probability weighting) aiming to explain unresponsiveness.

Result 3. *At the round-level, EV terms are correlated with lottery valuations that are responsive and are tightly concentrated around the expected value. In contrast, non-EV terms are correlated with valuations that are unresponsive and highly dispersed.*

The previous analysis has focused on lotteries. Appendix Figure A.2 shows the CDF of the valuations in the mirror tasks in the Calc treatment. Quantitatively similar results also appear for mirrors: The employment of EV terms in a mirror round predicts responsive and concentrated mirror valuations, while the employment of non-EV terms predicts the opposite.

5.2 Subject Types and Their Responsiveness

Do the calculator inputs of a subject predict their responsiveness? I categorize subjects by their *modal base terms* in lottery tasks in the Calc treatment. More specifically, for a subject, their modal base term is the base term that is used in the highest number of rounds, among all lottery rounds of this subject. Their *type* is the base term group to which their modal

[16] Appendix Figure A.1 displays the CDF of lottery valuations separated into three disjoint groups of rounds: 1) rounds employing only EV terms; 2) rounds employing only non-EV terms; and 3) rounds employing both EV and non-EV terms. The figure demonstrates that when valuations deviate from expected values despite the presence of EV terms, these deviations are primarily attributable to the concurrent use of non-EV terms rather than to the dropped numerical factors.

base term group belongs.[17] The types are a simple but powerful summary of the calculator inputs at the subject-level, due to the within-subject stability of calculations documented in Result 2. The five base term groups correspond to five subject types: 1) Expected value; 2) Number; 3) Linear box; 4) Linear money; 5) Non-linear.

To proceed with the analysis, I construct a quantitative measure of an individual subject's responsiveness using the regression slope of valuations on expected values. Specifically, I run the following regression

$$|\text{valuation}| = \alpha + r\ |\text{expected value}| + \epsilon \tag{1}$$

using all lottery tasks completed by a subject in the Calc treatment. The subject's *responsiveness* in lottery tasks is the estimated coefficient r. A responsiveness of 0 indicates complete unresponsiveness – on average, the valuations do not change with the expected values. On the other hand, a responsiveness of 1 indicates no unresponsiveness – the valuations change by the same magnitude as the expected values. This measure of an individual subject's responsiveness can be expanded to the mirror tasks and the NoCalc treatment by the slopes of analogous regressions.

Figure 5 shows the histograms of individual responsiveness in lottery tasks in the Calc treatment, first for all subjects and then separately for each type of subjects. The responsiveness is censored at a lower bound of -0.2 in the graphs for improved visibility. From the upper-left panel encompassing all subjects, it is immediate to see that responsiveness has a bimodal distribution – most subjects concentrate around no unresponsiveness (responsiveness = 1) and complete unresponsiveness (responsiveness = 0), and only a small fraction of subjects are in the middle range. The average responsiveness in the subject population is 0.48, showing that when the absolute EV of the lottery increases by one dollar, the valuation only increases by less than half a dollar.

The distributions of responsiveness for different subject types reveal substantial heterogeneity beneath the bimodal aggregate distribution. These two distinct modes correspond

[17] This categorization is based on base terms, as opposed to procedures, in order to incorporate all calculations made by a subject. The categorization is robust to using the modal procedure group for categorization, and also to a machine learning-based categorization utilizing the unsupervised topic model mentioned in Footnote 12. See Appendix B for more details for the latter.

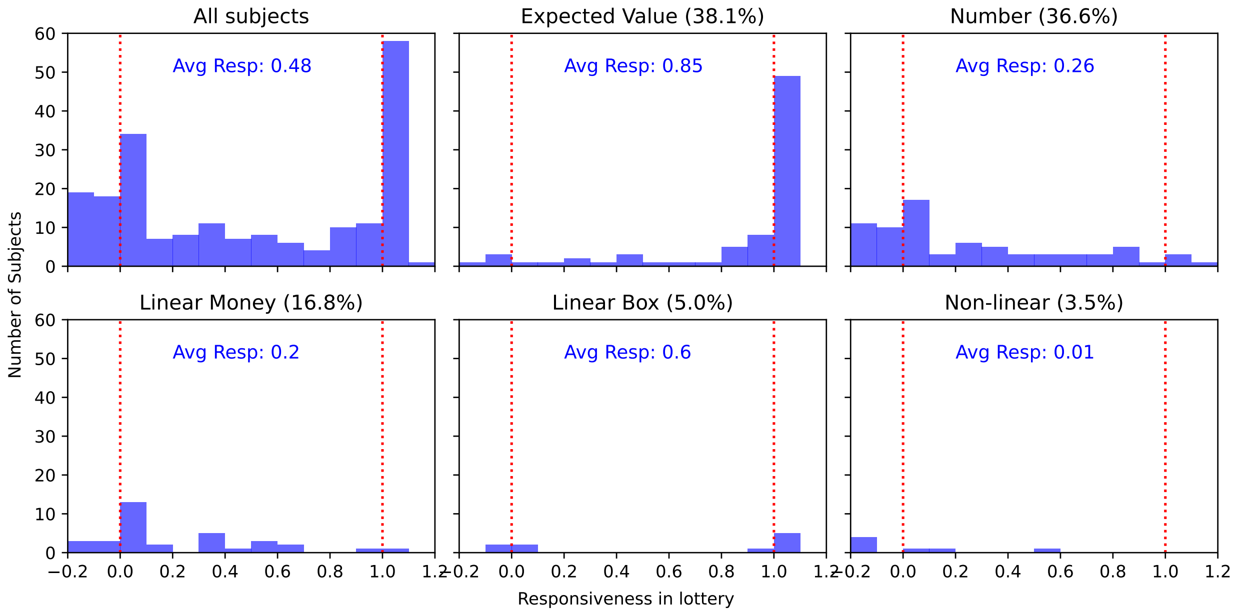


Figure 5: Histograms of individual unresponsiveness in lottery tasks in the Calc treatment

to two broad groups of subjects. The EV-type subjects (38.1% of the population) tend to exhibit responsiveness of nearly one (average = 0.85), which implies their lottery valuations closely track expected values. For example, these subjects value lotteries G8/L8 (with $|EV| = 2.08$) at an average of \$3.36, while valuing lotteries G92/L92 (with $|EV| = 23.92$) at an average of \$21.48. In contrast, non-EV-type subjects (61.9% of the population) exhibit much lower responsiveness (average = 0.25). Their valuations increase only minimally as probability increases: their average valuation of lotteries G8/L8 is \$11.4, while their average valuation of lotteries G92/L92 is only \$16.4, despite the large difference in expected values. In aggregate, this stark difference between EV-type and non-EV-type subjects produces the bimodal distribution of responsiveness observed in the overall sample.

Perhaps most strikingly, a substantial proportion of non-EV-type subjects exhibit responsiveness close to or even below zero. Among all non-EV-type subjects, 26.4% exhibit responsiveness between 0 and 0.1, and another 26.4% exhibit responsiveness below 0. For the non-EV-type subjects, their extreme unresponsiveness and prevalent voilations of monotonicity seriously challenge the genuine risk attitudes interpretation of unresponsiveness. Also note that the extreme unresponsiveness in the current data is far from being unique in the literature. For example, many subjects in Gonzalez and Wu (1999, function $w(\cdot)$ in Figure

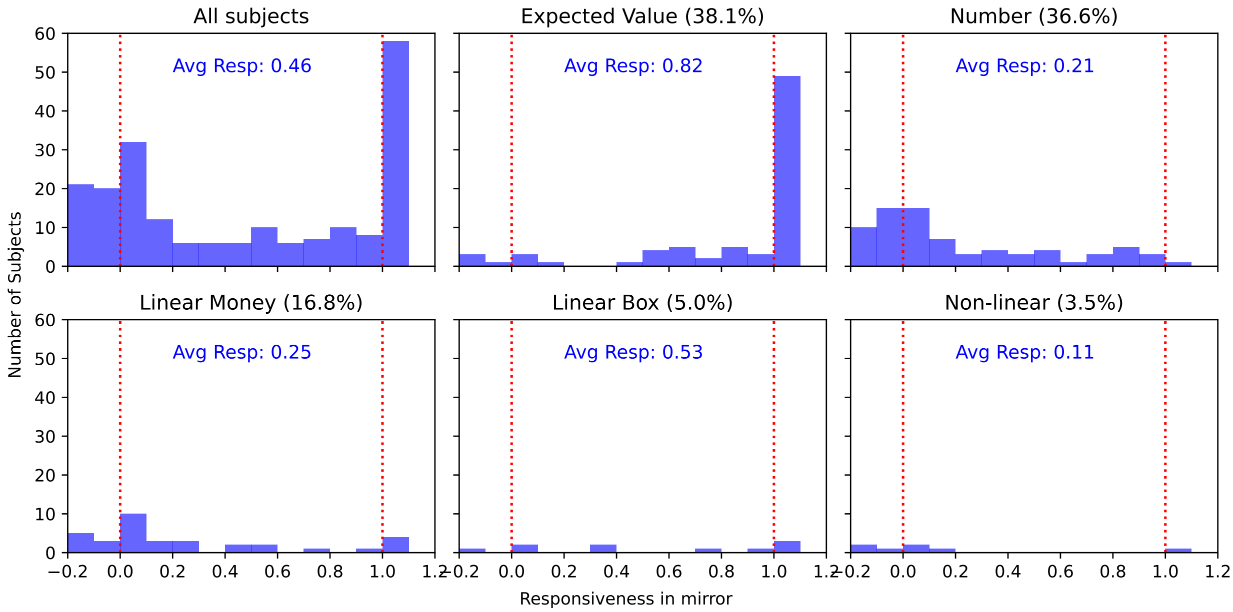


Figure 6: Histograms of individual unresponsiveness in mirror tasks in the Calc treatment

6, page 150) have very unresponsive probability weighting functions, especially when the probabilities are not too extreme.[18]

Moreover, although subject types are constructed only using calculator inputs in lottery tasks, they can also predict the responsiveness in mirror tasks. Figure 6 shows the histograms of individual responsiveness in mirror tasks in the Calc treatment, separately for each type of subjects. For all types, the responsiveness in mirror tasks is quantitatively similar to that in lottery tasks. The EV-type subjects submit responsive mirror valuations, but perhaps more interestingly, their mirror valuations are on average slightly *less* responsive than their lottery valuations. The non-EV-type subjects submit very unresponsive mirror valuations, and their mirror responsiveness is similar to that of lottery. These findings reinforce the roles played by complexity in the measured risk attitudes. On the one hand, although the EV-type subjects clearly know how to calculate the expected values, sometimes they do not do so even when valuing mirrors. On the other hand, the extreme unresponsiveness documented in the lottery

[18]Even though the subjects in Gonzalez and Wu (1999) are graduate students in psychology, who should be more responsive than my online sample (Benjamin, Brown and Shapiro, 2013), weak monotonicity is violated in 21% of the pairwise comparisons. Moreover, among their 10 subjects (excluding the one dropped from their main sample due to randomness and inconsistency of his responses), at least 3 subjects (Subjects 4, 7, and 8) have almost flat probablity weighting functions when the true probability is between 0.1 and 0.9. This proportion (3 out of 10, not considering the dropped Subject 11) is comparable with my data (32.7% of all subjects have responsiveness smaller than 0.1.).

valuations of the non-EV-type subjects is preserved in their mirror valuations.

Result 4. *In both lottery and mirror tasks, EV-type subjects tend to be very responsive, while non-EV-types subjects tend to be very unresponsive. In aggregate, the distribution of unresponsiveness is bimodal.*

The low responsiveness of the non-EV types are not entirely driven by those subjects who are inattentive and submit identical valuations across many tasks. In Appendix Figure A.3, I reproduce Figure 5 using all subjects excluding those who give the exact same absolute valuation to no fewer than 7 out of a total of 9 lottery tasks. The apparent bimodal distribution of responsiveness is robust to excluding these subjects, and the non-EV types still exhibit very low responsiveness.

5.3 Out-of-Sample Predictive Power of Calculator Inputs

Looking at Result 3 and Result 4, the Reader might be worried about the external validity of the patterns documented. Especially, whether the relationships between calculation inputs and valuations observed in the Calc treatment generalize to settings without the calculator? In order to answer this question, I examine the out-of-sample predictive power of calculator inputs. Specifically, I link the calculator inputs in the Calc treatment to the valuations in the NoCalc treatment, and show that the calculator inputs in the Calc treatment can also predict round-level and subject-level valuation patterns in the NoCalc treatment.

To examine the external validity of Result 3, for each round in the NoCalc treatment, I match it with the round in the Calc treatment where the same subject faces the same lottery task. I then reproduce Figure 4 using valuations from the NoCalc rounds and the calculator input from their matched Calc rounds. The reproduced graph can be found in Figure 7. The headline observations from Result 3 still hold. NoCalc rounds where the subjects employ an EV term in their matched Calc rounds generate responsive and concentrated lottery valuations, though they are not quite to the same degree as their matched Calc rounds.[19] Moreover, NoCalc rounds where the subjects employ any non-EV term in their matched

[19] K-S distance between G92/L92 and G8/L8: 0.61 in NoCalc vs. 0.94 in Calc; interquartile range for the four tasks: between 5 and 9.92 in NoCalc vs. always 0 in Calc.

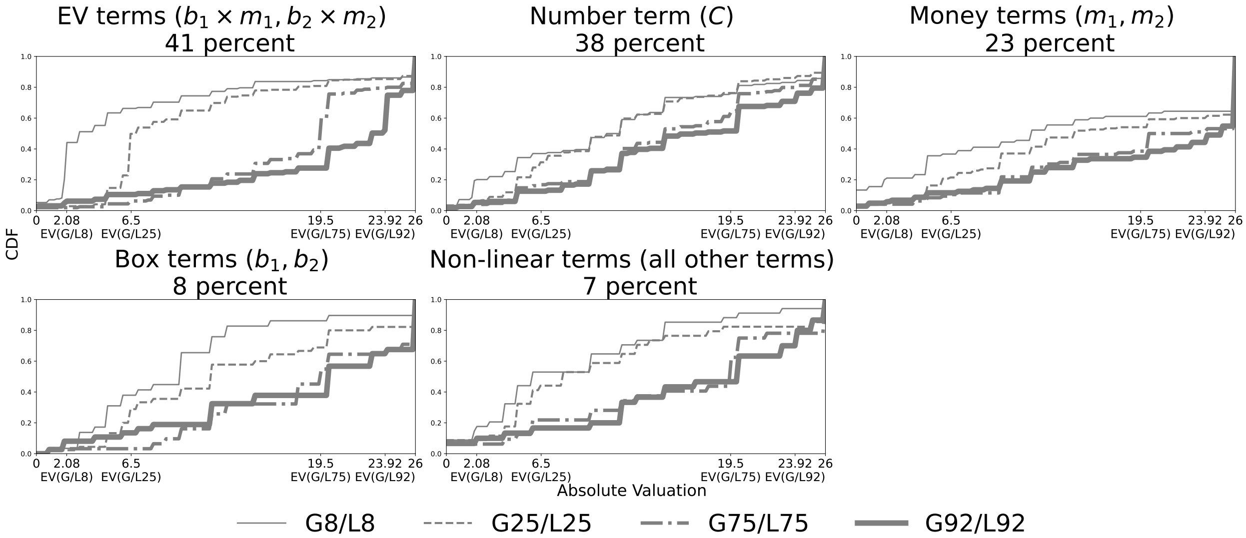


Figure 7: Distributions of lottery valuations in the NoCalc treatment, grouped by the base terms used by the same subjects in the corresponding Calc treatment rounds (see the text for details)

Calc rounds again generate unresponsive and dispersed lottery valuations. The quantitative measures of K-S distance and interquartile range reveal that the valuations in these NoCalc rounds are roughly as unresponsive and dispersed as those in their matched Calc rounds. Some of the idiosyncratic patterns conditional on groups are also preserved. For example, conditional on the matched Calc round using a money base term, the NoCalc valuations again have large point masses at \$26 (the maximum absolute valuation).

Figure 8 shows the distributions of individual responsiveness in lottery tasks in the NoCalc treatment. First, even though types are constructed only using calculator input data from the Calc treatment, they are still predictive of the responsiveness in the NoCalc treatment – the EV-types exhibits higher responsiveness than the non-EV types in the NoCalc treatment. Second, the aggregate distribution of responsiveness in the subject population differs between treatments – the NoCalc treatment shows a less pronounced bimodal pattern than the Calc treatment. There are relatively fewer subjects with high responsiveness (near 1) and relatively more subjects with intermediate or low responsiveness (near 0) in NoCalc compared to Calc. Third, this difference in distributions between treatments is attributable solely to the EV types. The EV types exhibits moderately lower responsiveness in the NoCalc treatment when compared with their performance in the Calc treatment, while the other types show

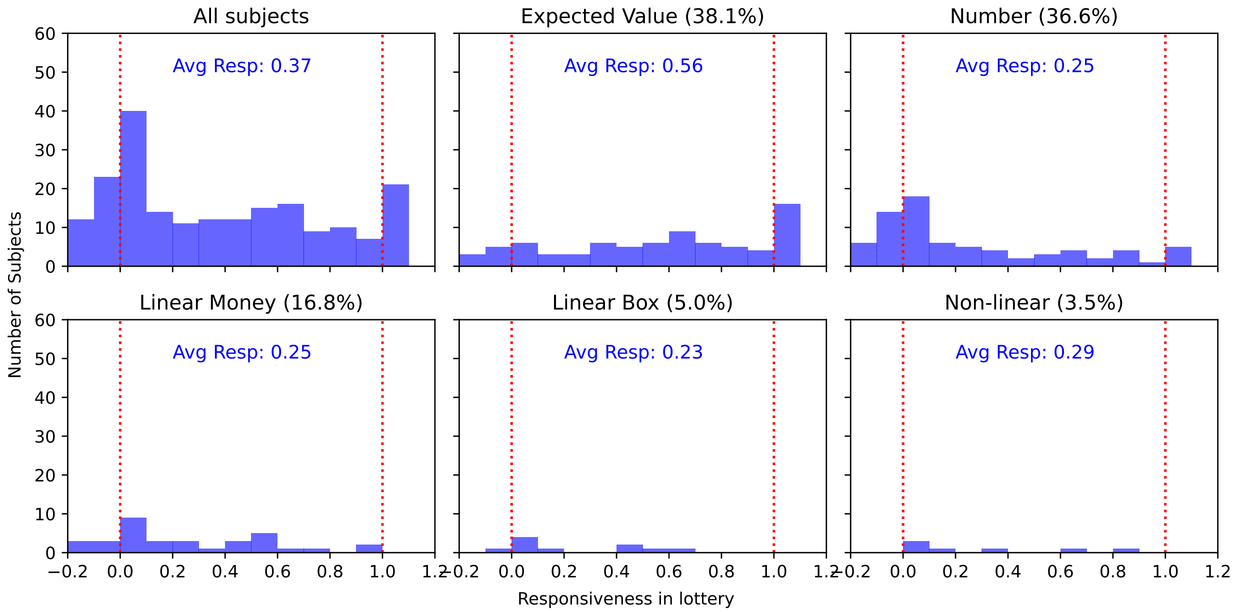


Figure 8: Histograms of individual responsiveness in lottery tasks in the NoCalc treatment

statistically similar responsiveness across treatments. Using the Wilcoxon signed-rank test to compare the responsiveness in the two treatments separately for each subject type confirms this result – The Wilcoxon test rejects the null hypothesis of equal between-treatments responsiveness for the EV types, but cannot reject it for any other subject type.[20]

Result 5. *Calculator inputs predict lottery valuations in the NoCalc treatment. EV-type subjects (as identified in the Calc treatment) exhibit significantly higher responsiveness in NoCalc than non-EV types, maintaining the qualitative relationship observed in the Calc treatment.*

Result 6. *Comparing across treatments reveals an overall shift toward higher unresponsiveness in the NoCalc treatment. This shift is entirely driven by EV-type subjects, who exhibit moderately higher unresponsiveness in NoCalc than in Calc. In contrast, non-EV types show statistically indistinguishable levels of unresponsiveness across NoCalc and Calc treatments.*

[20]The Wilcoxon test p-value for linear box-type subjects is 0.049, which is statistically insignificant after controlling for familywise error rate.

6 Mechanisms behind Unresponsiveness

What mechanisms explain the unresponsiveness observed in lottery valuations? One of the key advantages of my experimental design is that I collect detailed calculator input data, which can help distinguish between theoretical mechanisms that are indistinguishable using only standard lottery valuation data. This section uses this calculator input data to examine the mechanisms behind unresponsive lottery valuations from two perspectives.

First, to the extent that a broad notion of complexity can explain unresponsiveness, the exact nature of this complexity remains under debate. I classify complexity into one of two camps, and seek evidence in the data that supports each camp. The *implementation* camp attributes unresponsiveness to costs of implementing optimal procedures despite awareness of them, while the *representation* camp attributes it to fundamental misconceptions about the valuation problem itself.[21] The distinctions between the two camps are of fundamental interest – if the first camp prevails, unresponsive lottery valuations documented in the lab experiments will have limited predictive power for high-stakes real life decisions under risk, since the implementation costs can be overcome with the high-stakes, while the latter camp indicates the opposite, and calls for policy solutions to help people make decisions under uncertainty.

Second, I leverage the calculator input data to directly benchmark observed calculations against key theoretical models of probability weighting and cognitive imprecision. I interpret these models literally, as descriptions of actual calculation steps rather than "as-if" models. Using this approach, I derive the specific calculations predicted by each theoretical model. I acknowledge the caveats of interpreting theoretical models literally. Nevertheless, by comparing the predictions of these models to the observed calculation data, I can assess how accurately standard theoretical models describe the actual calculations employed in lottery valuations.

[21]These two camps are similar to the "frictions" and "mental gaps" camps in Handel and Schwartzstein (2018).

6.1 Implementation and Representation Complexity

Implementation Complexity The *implementation* camp of complexity explains unresponsive lottery valuations by the costs of implementing valuation procedures that represent their genuine risk attitudes (Payne, Bettman and Johnson, 1988, 1993, Oprea, 2024*a*). For example, it might be costly for economic agents to calculate the expected values of lotteries. As a result, even if some agents are risk neutral and understand that calculating the expected values leads to their preferred lottery valuations, these subjects may opt into using less costly calculations and, in turn, submit lottery valuations that do not fully reflect their genuine risk attitudes. When valuing mirrors, these agents face analogous implementation costs in calculating the expected values, and may similarly choose simpler calculations that generate valuations similar to those of lotteries.

This complexity camp offers two predictions that are relevant to the current study. First, decreasing the implementation costs should lead to more responsive valuations. This prediction is verified when I compare the valuations in the NoCalc and Calc treatments for the EV-type subjects. Specifically, in Result 6, I document that the EV-type subjects submit much more responsive valuations in the Calc treatment, where implementation costs are arguably smaller, than in the NoCalc treatment.[22] This finding has important implications for interpreting previous studies, as the NoCalc treatment closely resembles the traditional approach to eliciting lottery valuations used in the literature. Since EV-type subjects submit less than fully responsive lottery valuations in the NoCalc treatment due to implementation complexity, it is reasonable to infer that the same mechanism has contributed to the unresponsiveness observed in past studies, which similarly did not provide explicit calculation tools. In other words, had these subjects in other studies had ready access to a calculator, the past studies will see many more of their subjects exhibiting risk neutrality when valuing lotteries.

A second prediction from of implementation complexity is that the calculations generating unresponsive valuations should be less costly than those generating responsive valuations.

[22] Moreover, in the reconciliation stage where the subjects are given a chance to reconcile their inconsistent valuations in the NoCalc and Calc stages for the same task, these EV-type subjects prefer their valuation in the Calc treatment more than twice as frequently as they prefer their valuation in the NoCalc treatment. In other words, the almost risk-neutral valuations submitted by these subjects in Calc better represent their genuine risk attitudes than the ones submitted in NoCalc. See Appendix E for more details.

Task Type	Subject Type	Mean Length	Median Length
lottery	expected value	11.3	11
	number	4.4	3
	linear money	4.5	4
	linear box	11.0	10
	nonlinear	13.0	12
mirror	expected value	11.5	11
	number	5.4	3
	linear money	5.1	4
	linear box	9.9	10
	nonlinear	12.7	11

Table 6: Length of Calculator Inputs (in Characters) by Subject Type and Task Type

This prediction can be tested using the calculator input data collected in the Calc treatment. I measure the costs of the calculations using a simple metric: the *length* of calculator inputs, which is defined as the total number of characters (digits, operation signs, and decimal points) in all numerical expressions in the calculator input. This metric directly captures implementation costs in our experimental environment since, by design of the calculator, each character requires a separate operation – either a button click or a keystroke – to type. For example, calculating $75 \times 26/100$ requires nine separate operations, while entering 15 requires only two.

Table 6 shows the mean and median length of calculator inputs by subject type and task type. The data reveals a clear pattern: number-type and linear money-type subjects consistently perform less costly calculations than the EV types, for both lottery and mirror tasks. This aligns with the predictions of implementation complexity. However, the current experimental design cannot establish a *causal* link between implementation costs and the choice of shorter calculator inputs, and in turn, the unresponsive valuations they generate.

To summarize these findings, I provide strong evidence that implementation complexity plays a significant role in the unresponsiveness exhibited by EV-type subjects in the NoCalc treatment. For number and linear money types, the evidence supporting the implementation complexity explanation is suggestive though less definitive.

Representation Complexity The *representation* camp of complexity focuses on incorrect mental representations of the valuation task. Specifically in the current context, having incorrect mental representations refers to the scenario where the agents lack the knowledge to aggregate disaggregated monetary outcomes of a lottery. As a result, when asked to provide valuations of lotteries, these subjects submit valuations that do not represent their genuine risk attitudes. Since the lack of knowledge to aggregate is also present in the absence of risk, it should also lead to similar valuations of mirrors.

Using the calculator input data, I find two patterns that are strong indications of representation complexity. First, looking at Table 6, the linear box and nonlinear type subjects use calculator inputs that are as long as or even longer than those of the EV-type. This supports the presence of representation complexity, as these subjects willingly perform lengthy calculations yet still exhibit strong unresponsiveness, and thus can't be explained by saving implementation costs. To the extent that complexity is behind their calculations and valuations, representation complexity is consistent with these subjects.

Second, I can identify a subset of valuations that are very likely consequences of representation complexity from the calculations. This identification process uses the feature of procedures defined in Section 4.3, which capture the functional form that the subjects use to map the primitives of the task to their valuations. If a procedure is strictly decreasing in any of the monetary outcomes, the procedure would imply a lower valuation when the monetary outcome increases. Similarly, if a procedure is a strictly decreasing function in b_1 (the number of boxes containing a non-zero amount of money), the procedure would imply a lower (absolute) valuation when the probability of the non-zero outcome increases. These two scenarios, when observed, strongly suggest that the observed valuations are due to representation complexity. Formally, I define a procedure to be *decreasing* if the procedure (after substituting b_2 with $100 - b_1$) is strictly decreasing in any of the primitives m_1, m_2, or b_1, at any point within the range $\{(m_1, m_2, b_1) : m_1 \geq 0, m_2 \geq 0, 0 \leq b_1 \leq 100\}$. For example, among all procedures listed in Panel A of Table 4, there are two decreasing procedures: $b_2 \times m_1/100$ and $100/b_1$.

Table 7 shows the shares of decreasing procedures for each subject type, separately for the two task types. Nonlinear type subjects disproportionately use decreasing procedures in

Task Type	Subject Type	%decreasing
lottery	expected value	2.2%
	number	3.3%
	linear box	2.5%
	linear money	5.0%
	nonlinear	39.9%
mirror	expected value	2.2%
	number	3.3%
	linear box	16.7%
	linear money	6.1%
	nonlinear	37.9%

Table 7: Share of decreasing procedures by subject type and task type

both lottery and mirror tasks, while other types mostly avoid using decreasing procedures. It is important to note that while the presence of a decreasing procedure strongly suggests the presence of representation complexity (sufficiency), the absence of a decreasing procedure does not meaningfully suggest the absence of representation complexity (necessity). For example, a number procedure is by definition never decreasing, since none of the primitives appears in the procedure.[23] But a number procedure may still be a result from representation complexity.

In summary, I provide strong evidence in support of the presence of representation complexity in the non-linear type subjects, since these subjects disproportionately use decreasing procedures and lengthy calculator inputs. The calculator input lengths are also supportive of representation complexity's presence in the linear box type subjects.

6.2 Probability Weighting and Cognitive Imprecision

Turning away from the two broad camps of complexity, I now examine two specific mechanisms – probability weighting and cognitive imprecision – that have been proposed in the behavioral economic literature to explain unresponsiveness in lottery valuations.

[23]Though number (and expected value) procedures are by definition not decreasing, since Table 7 shows the shares of decreasing procedures by *subject* type instead of *procedure* group, the shares of decreasing procedure among these subject types can still be positive.

(Naive) Probability Weighting Naive probability weighting occurs when economic agents evaluate lotteries by applying subjective probability weights to monetary outcomes rather than using the objective probabilities. Formally, these weights are determined by a probability weighting function $w(\cdot)$ that transforms objective probabilities into probability weights. In the current context, for a lottery Gn, the valuation generated by naive probability weighting would be $w(n/100) \times 26$, where $w(\cdot)$ is the probability weighting function, and analogously for a lottery Ln.

If subjects literally implement this probability weighting process, it should generate specific, testable patterns in their calculator inputs. Most importantly, naive probability weighting predicts that subjects would use linear money procedures in their calculations. To illustrate, consider an agent valuing (G75, Lottery) using naive probability weighting. This agent would calculate their valuation as $w(0.75) \times 26$. If, for example, $w(0.75) = 0.6$, the agent would compute 0.6×26. From the experimenter's perspective, this calculation would appear as a linear money procedure – specifically, one that multiplies the monetary amount (26) by a coefficient (0.6) that differs from the true probability (0.75) and cannot be matched with any task primitives. More generally, naive probability weighting will always generate linear money procedures, because the transformed probability weights become the coefficients in linear combinations of monetary amounts.

The tractable structure of linear money procedures allows for direct recovery of subjects' probability weighting function, assuming these procedures represent literal probability weighting behavior. By examining the coefficients in these linear money procedures, I can directly recover each subject's probability weights for different probabilities. Specifically, when a subject valuing a lottery Gn/Ln uses a linear money procedure with coefficient γ_1 on m_1, this reveals that their probability weight would be $w(n/100) = \gamma_1$ under this interpretative framework. To give a substantive example, if a subject uses the procedure $0.6 \times m_1$ when valuing (G75, Lottery), I can infer that the probability weight is $w(0.75) = 0.6$. More generally, when a subject uses a linear money procedure $\gamma_0 + \gamma_1 \times m_1 + \gamma_2 \times m_2$, the coefficient γ_1 reveals the probability weight placed on the outcome m_1.[24] This recovered weighting

[24] I only use γ_1, not γ_2 to recover the probability weighting function because m_2 is always zero in our experimental design. As a result, subjects may rationally omit terms involving m_2 from their calculations, making it impossible to reliably recover the probability weights placed on m_2.

function is meaningful specifically in the context of testing probability weighting as a literal decision-making process.

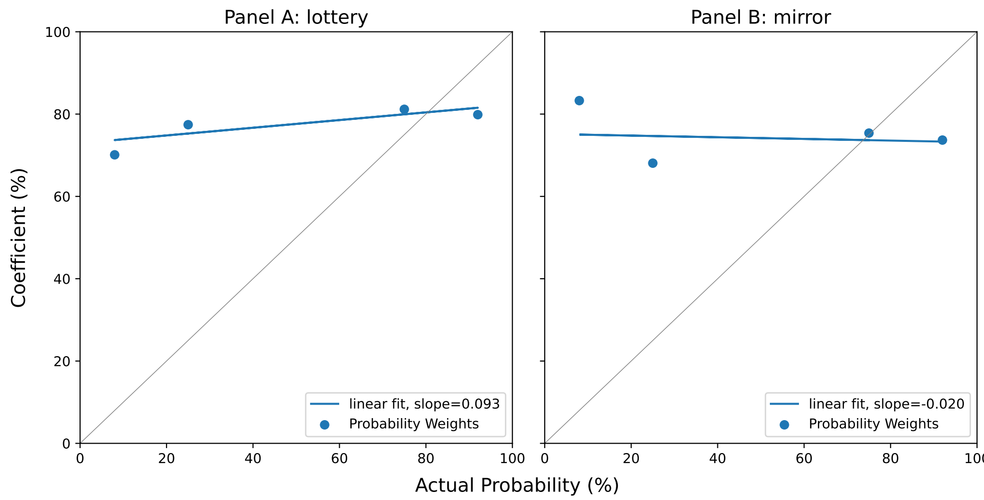


Figure 9: Coefficients recovered from linear money procedures for linear money subjects. These coefficients can be interpreted either as probability weights under probability weighting theory or as perceived probabilities under cognitive imprecision theory, depending on which framework is applied.

Panel A of Figure 9 displays the recovered probability weighting function as the average of the recovered probability weights. The sample is restricted to the rounds where linear money procedures are used by linear money-type subjects.[25] The extremely low slope (0.093) of the probability weighting function demonstrates that the probability weights are almost entirely unresponsive to changes in the actual probability. Such extreme unresponsiveness is difficult to reconcile with probability weighting as a theoretical account of these subjects' behavior. Instead, the pattern is more consistent with subjects employing simplified heuristics such as attribute substitution (Kahneman and Frederick, 2002, Fan, Liang and Peng, 2024), rather than expressing genuine probability weighting. Moreover, the probability weighting function recovered using the calculator input data presents a stark contrast to the inverse S-shaped probability weighting function typically recovered using valuation data (e.g., Camerer and Ho, 1994, Gonzalez and Wu, 1999). Thus, while linear money subjects' use of linear

[25] I restrict the attention to the linear money type because the other types use linear money terms only inconsistently. This inconsistency is contradictory to the prediction of cognitive imprecision that linear procedures, and by extension linear money terms, should be *consistently* used. Lifting this restriction leads to qualitatively similar results.

money procedures superficially aligns with probability weighting's predictions, evidence from the calculator inputs casts significant doubt on probability weighting as the underlying mechanism generating their behavior, at least when interpreted as a literal description of the decision-making process.

Cognitive Imprecision Khaw, Li and Woodford (2021, KLW) propose that cognitive imprecision can explain unresponsiveness in lottery valuations. In their model, the agent makes decisions based not on the exact primitives of the decision tasks, but rather on perceived values of the primitives that may systematically deviate from the true values. For a lottery Gn, the valuation generated by cognitive imprecision would be $\pi(n/100) \times 26$, where $\pi(\cdot)$ represents the perceived probability function, and analogously for a lottery Ln. Cognitive imprecision has been viewed as one of the most promising avenues to explain unresponsive lottery valuations (Woodford, 2012, Frydman and Jin, 2021, Khaw, Li and Woodford, 2021, Barretto-García et al., 2023, Vieider, 2024, Oprea and Vieider, 2024). Importantly, since cognitive imprecision affects the perception of numerical primitives regardless of whether it appears in lotteries or mirrors, the theory predicts similar patterns of unresponsiveness across both lottery and mirror tasks. This characteristic allows it to explain the intriguing patterns of similar lottery and mirror valuations in Oprea (2024*b*).

Taking KLW's model as a literal description of the decision-making process, the model again predicts the use of linear money procedures, for the same reason as probability weighting: subjects multiply the monetary amount by a coefficient representing their subjective perception. Through the same logic used in the previous section, I can recover the average perceived probability from the coefficients in linear money procedures.

Figure 9 shows that the recovered perceived probabilities suffer from extreme unresponsiveness to changes in the actual probability, and the slope is even negative in mirrors. This extreme unresponsiveness is difficult to reconcile with cognitive imprecision, which predicts that perceived probabilities should at least partially track actual probabilities, with a bias toward the prior. Furthermore, the absolute values of these perceived probabilities are problematically high, especially for small probabilities. According to KLW's model, cognitive imprecision should bias perceived probabilities toward the prior mean (typically in the middle of the range, around 50%), yet we observe perceived probabilities of approximately 70% even for actual probabilities as low as 8% and 25%. This combination of extreme unresponsiveness and unexpectedly high values suggests that subjects using linear money procedures likely employ simpler heuristics unrelated to cognitive imprecision.

7 Connections to the Literature

This study makes contributions to a few strands of literature. First, it contributes to the literature that studies the roles played by cognitive complexity in the measured risk attitudes. The literature has amassed significant evidence that, at least to some extent, the observed departure of small-stake lottery valuations from their expected values are consequences of a broad notion of cognitive complexity, as opposed to fully reflecting the genuine risk attitudes (Oprea, 2024*b*, Payne, Bettman and Johnson, 1988, Harbaugh, Krause and Vesterlund, 2010, Woodford, 2012, Benjamin, Brown and Shapiro, 2013, Pachur et al., 2013, Deck and Jahedi, 2015, Pachur et al., 2018, Martínez-Marquina, Niederle and Vespa, 2019, Khaw, Li and Woodford, 2021, Frydman and Jin, 2021, Nielsen and Rehbeck, 2022, Barretto-García et al., 2023, Enke and Graeber, 2023, Enke and Shubatt, 2023, McGranaghan et al., 2024, Puri, 2025). As to the exact nature of this complexity, the extant literature has proposed a few candidate sources, but has not yet reached a consensus over the most importance sources of complexity. My main contribution to this literature is twofold. On the one hand, by revealing subjects' calculations behind the elicited lottery valuations and linking the calculations to measured risk attitudes, I shed light on the sources of this complexity. On the other hand, I show the same subject's calculations are generally similar for lotteries and mirrors, emphasizing the roles played by complexity in the measured risk attitudes from a procedural perspective.

Second, the study contributes to the interdisciplinary literature that measures the decision-making processes that generate observed choices. To reveal this usually unobserved layer, studies in this literature use various techniques including mouse tracking (Payne, Bettman and Johnson, 1988), eye tracking (Reutskaja et al., 2011, Brocas et al., 2014), intermediate choice tracking (Caplin, Dean and Martin, 2011), belief measurement (Aoyagi, Fréchette and Yuksel, 2024), and verbal descriptions of decision-making process (Ericsson and Simon, 1980, Kendall and Oprea, 2023, Arrieta and Nielsen, 2023). Schulte-Mecklenbeck et al. (2017) provides a recent review of these techniques. Most relevantly, a branch of this literature has applied these techniques to risk attitudes, the very question studied here (Payne, Bettman and Johnson, 1988, Arieli, Ben-Ami and Rubinstein, 2011, Pachur et al., 2013, 2018, Arrieta and Nielsen, 2023). The techniques developed so far have mostly focused on the information acquisition aspect of the decision-making process, that is, what information is accessed. This study makes an important methodological contribution to this literature by developing the calculator design that recovers the computational aspect of the decision-making process, i.e., how decision-makers use the accessed information to perform calculations, which cannot be recovered by previous techniques. The calculator design developed in this study, along with the features of procedures and base terms, can be easily transplanted and deployed to studying other topics.

This study also contributes to the literature that explicitly models and measures procedural decision-making (Simon, 1955, Payne, Bettman and Johnson, 1988, 1993, Oprea, 2020, Banovetz and Oprea, 2023, Arrieta and Nielsen, 2023, Oprea, 2024*a*). This literature takes procedures as the fundamental object that economic decision-makers need to choose in the decision-making processes. In other words, while the standard theory aims to describe how people choose from feasible *actions*, this literature aims to describe how people choose from feasible *procedures*, each is a mapping from the task primitives to the actions. A particular focus of this literature is how the characteristics of the decision-making environment and the implementation costs of these procedures affect the use of these procedures and the actions resulting from these procedures. The current study records the computational aspect of the procedures and measures the implementation costs, and thus provides direct tests of the predictions made by this literature.

More broadly, this study joins a long list of literature studying anomalies in risky choices (e.g., Kahneman and Tversky, 1979, 1984, Tversky and Kahneman, 1992, Camerer and Ho, 1994, Hey and Orme, 1994, Prelec, 1998, Wu and Gonzalez, 1996, Gonzalez and Wu, 1999, Blavatskyy, 2007, Wakker, 2010, Barberis, 2013, Barseghyan et al., 2013, Bhargava, Loewenstein and Sydnor, 2017, O'Donoghue and Somerville, 2018, Oprea, 2024*b*, McGranaghan et al., 2024). This study micro-founds the observed unresponsiveness and fourfold patterns in lottery valuations by documenting the decision-making processes behind these valuations.

References

Aoyagi, Masaki, Guillaume R. Fréchette, and Sevgi Yuksel. 2024. “Beliefs in Repeated Games: An Experiment.” *American Economic Review*, 114(12): 3944–3975.

Arieli, Amos, Yaniv Ben-Ami, and Ariel Rubinstein. 2011. “Tracking Decision Makers under Uncertainty.” *American Economic Journal: Microeconomics*, 3(4): 68–76.

Arrieta, Gonzalo, and Kirby Nielsen. 2023. “Procedural Decision-Making In The Face Of Complexity.”

Banki, Daniel, Uri Simonsohn, Robert Walatka, and George Wu. 2025. “Decisions under Risk Are Decisions under Complexity: Comment.”

Banovetz, James, and Ryan Oprea. 2023. “Complexity and Procedural Choice.” *American Economic Journal: Microeconomics*, 15(2): 384–413.

Barberis, Nicholas C. 2013. “Thirty Years of Prospect Theory in Economics: A Review and Assessment.” *Journal of Economic Perspectives*, 27(1): 173–196.

Barretto-García, Miguel, Gilles de Hollander, Marcus Grueschow, Rafael Polanía, Michael Woodford, and Christian C. Ruff. 2023. “Individual Risk Attitudes Arise from Noise in Neurocognitive Magnitude Representations.” *Nature Human Behaviour*, 7(9): 1551–1567.

Barseghyan, Levon, Francesca Molinari, Ted O’Donoghue, and Joshua C. Teitelbaum. 2013. “The Nature of Risk Preferences: Evidence from Insurance Choices.” *American Economic Review*, 103(6): 2499–2529.

Becker, Gordon M., Morris H. Degroot, and Jacob Marschak. 1964. “Measuring Utility by a Single-Response Sequential Method.” *Behavioral Science*, 9(3): 226–232.

Benjamin, Daniel J., Sebastian A. Brown, and Jesse M. Shapiro. 2013. “Who Is ’Behavioral’? Cognitive Ability and Anomalous Preferences.” *Journal of the European Economic Association*, 11(6): 1231–1255.

Bhargava, Saurabh, George Loewenstein, and Justin Sydnor. 2017. “Choose to Lose: Health Plan Choices from a Menu with Dominated Option*.” *The Quarterly Journal of Economics*, 132(3): 1319–1372.

Blavatskyy, Pavlo R. 2007. “Stochastic Expected Utility Theory.” *Journal of Risk and Uncertainty*, 34(3): 259–286.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. “Latent Dirichlet Allocation.” *J. Mach. Learn. Res.*, 3(null): 993–1022.

Brocas, Isabelle, Juan D. Carrillo, Stephanie W. Wang, and Colin F. Camerer. 2014. "Imperfect Choice or Imperfect Attention? Understanding Strategic Thinking in Private Information Games." *The Review of Economic Studies*, 81(3): 944–970.

Camerer, Colin F., and Teck-Hua Ho. 1994. "Violations of the Betweenness Axiom and Nonlinearity in Probability." *Journal of Risk and Uncertainty*, 8(2): 167–196.

Caplin, Andrew, Mark Dean, and Daniel Martin. 2011. "Search and Satisficing." *American Economic Review*, 101(7): 2899–2922.

Chen, Daniel L., Martin Schonger, and Chris Wickens. 2016. "oTree—An Open-Source Platform for Laboratory, Online, and Field Experiments." *Journal of Behavioral and Experimental Finance*, 9: 88–97.

Deck, Cary, and Salar Jahedi. 2015. "The Effect of Cognitive Load on Economic Decision Making: A Survey and New Experiments." *European Economic Review*, 78: 97–119.

Enke, Benjamin, and Cassidy Shubatt. 2023. "Quantifying Lottery Choice Complexity." National Bureau of Economic Research w31677, Cambridge, MA.

Enke, Benjamin, and Thomas Graeber. 2023. "Cognitive Uncertainty*." *The Quarterly Journal of Economics*, 138(4): 2021–2067.

Enke, Benjamin, Thomas Graeber, Ryan Oprea, and Jeffrey Yang. 2024. "Behavioral Attenuation." National Bureau of Economic Research w32973, Cambridge, MA.

Ericsson, K. Anders, and Herbert A. Simon. 1980. "Verbal Reports as Data." *Psychological Review*, 87(3): 215–251.

Fan, Tony Q., Yucheng Liang, and Cameron Peng. 2024. "The Inference-Forecast Gap in Belief Updating."

Frydman, Cary, and Lawrence J Jin. 2021. "Efficient Coding and Risky Choice." *The Quarterly Journal of Economics*, 137(1): 161–213.

Gonzalez, Richard, and George Wu. 1999. "On the Shape of the Probability Weighting Function." *Cognitive Psychology*, 38(1): 129–166.

Handel, Benjamin, and Joshua Schwartzstein. 2018. "Frictions or Mental Gaps: What's Behind the Information We (Don't) Use and When Do We Care?" *Journal of Economic Perspectives*, 32(1): 155–178.

Harbaugh, William T, Kate Krause, and Lise Vesterlund. 2010. "The Fourfold Pattern of Risk Attitudes in Choice and Pricing Tasks." *The Economic Journal*, 120(545): 595–611.

Hey, John D., and Chris Orme. 1994. "Investigating Generalizations of Expected Utility Theory Using Experimental Data." *Econometrica*, 62(6): 1291–1326.

Kahneman, Daniel, and Amos Tversky. 1979. "Prospect Theory: An Analysis of Decision under Risk." *Econometrica*, 47(2): 263–291.

Kahneman, Daniel, and Amos Tversky. 1984. "Choices, Values, and Frames." *American Psychologist*, 39(4): 341–350.

Kahneman, Daniel, and Shane Frederick. 2002. "Representativeness Revisited: Attribute Substitution in Intuitive Judgment." In *Heuristics and Biases: The Psychology of Intuitive Judgment.* 49–81. New York, NY, US:Cambridge University Press.

Kendall, Chad, and Ryan Oprea. 2023. "On the Complexity of Forming Mental Models." *Quantitative Economics.*

Khaw, Mel Win, Ziang Li, and Michael Woodford. 2021. "Cognitive Imprecision and Small-Stakes Risk Aversion." *The Review of Economic Studies*, 88(4): 1979–2013.

l'Haridon, Olivier, and Ferdinand M. Vieider. 2019. "All over the Map: A Worldwide Comparison of Risk Preferences." *Quantitative Economics*, 10(1): 185–215.

Martínez-Marquina, Alejandro, Muriel Niederle, and Emanuel Vespa. 2019. "Failures in Contingent Reasoning: The Role of Uncertainty." *American Economic Review*, 109(10): 3437–3474.

McGranaghan, Christina, Kirby Nielsen, Ted O'Donoghue, Jason Somerville, and Charles D. Sprenger. 2024. "Distinguishing Common Ratio Preferences from Common Ratio Effects Using Paired Valuation Tasks." *American Economic Review*, 114(2): 307–347.

Nielsen, Kirby, and John Rehbeck. 2022. "When Choices Are Mistakes." *American Economic Review*, 112(7): 2237–2268.

O'Donoghue, Ted, and Jason Somerville. 2018. "Modeling Risk Aversion in Economics." *Journal of Economic Perspectives*, 32(2): 91–114.

Oprea, Ryan. 2020. "What Makes a Rule Complex?" *American Economic Review*, 110(12): 3913–3951.

Oprea, Ryan. 2024*a*. "Complexity and Its Measurement."

Oprea, Ryan. 2024*b*. "Decisions under Risk Are Decisions under Complexity." *American Economic Review*, 114(12): 3789–3811.

Oprea, Ryan, and Ferdinand M Vieider. 2024. "Minding the Gap: On the Origins of Probability Weighting and the Description-Experience Gap."

Pachur, Thorsten, Michael Schulte-Mecklenbeck, Ryan O Murphy, and Ralph Hertwig. 2018. "Prospect Theory Reflects Selective Allocation of Attention." *Journal of Experimental Psychology: General.*

Pachur, Thorsten, Ralph Hertwig, Gerd Gigerenzer, and Eduard Brandstätter. 2013. "Testing Process Predictions of Models of Risky Choice: A Quantitative Model Comparison Approach." *Frontiers in Psychology*, 4.

Payne, John W., James R. Bettman, and Eric J. Johnson. 1988. "Adaptive Strategy Selection in Decision Making." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(3): 534–552.

Payne, John W., James R. Bettman, and Eric J. Johnson. 1993. *The Adaptive Decision Maker.* Cambridge:Cambridge University Press.

Prelec, Drazen. 1998. "The Probability Weighting Function." *Econometrica*, 66(3): 497–527.

Puri, Indira. 2025. "Simplicity and Risk." *The Journal of Finance*, 80(2): 1029–1080.

Reutskaja, Elena, Rosemarie Nagel, Colin F. Camerer, and Antonio Rangel. 2011. "Search Dynamics in Consumer Choice under Time Pressure: An Eye-Tracking Study." *American Economic Review*, 101(2): 900–926.

Schulte-Mecklenbeck, Michael, Joseph G. Johnson, Ulf Böckenholt, Daniel G. Goldstein, J. Edward Russo, Nicolette J. Sullivan, and Martijn C. Willemsen. 2017. "Process-Tracing Methods in Decision Making: On Growing Up in the 70s." *Current Directions in Psychological Science*, 26(5): 442–450.

Simon, Herbert A. 1955. "A Behavioral Model of Rational Choice." *The Quarterly Journal of Economics*, 69(1): 99–118.

Tversky, Amos, and Daniel Kahneman. 1992. "Advances in Prospect Theory: Cumulative Representation of Uncertainty." *Journal of Risk and Uncertainty*, 5(4): 297–323.

Vieider, Ferdinand M. 2024. "Decisions Under Uncertainty as Bayesian Inference on Choice Options." *Management Science*, 70(12): 9014–9030.

Wakker, Peter P. 2010. *Prospect Theory: For Risk and Ambiguity.* Cambridge:Cambridge University Press.

Woodford, Michael. 2012. "Inattentive Valuation and Reference-Dependent Choice."

Wu, George. 2025. "There Is No Smoke with Mirrors When Instructions Are Clear."

Wu, George, and Richard Gonzalez. 1996. "Curvature of the Probability Weighting Function." *Management Science*, 42(12): 1676–1690.

Appendices

A Additional Figures and Tables

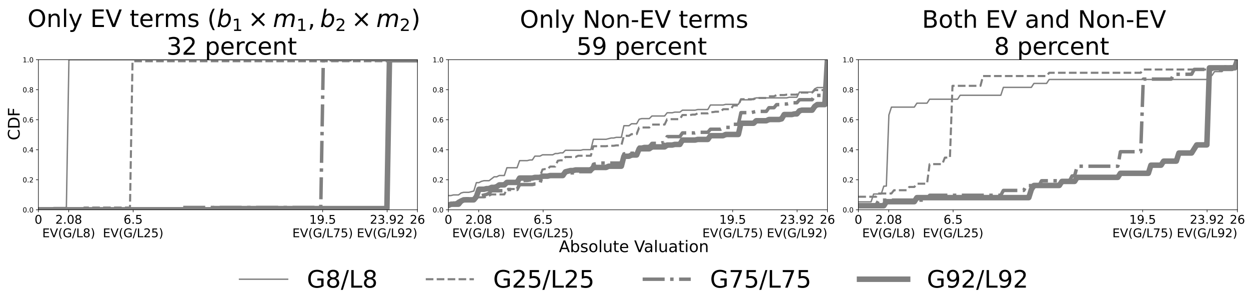


Figure A.1: CDF of lottery valuations for three disjoint groups of rounds

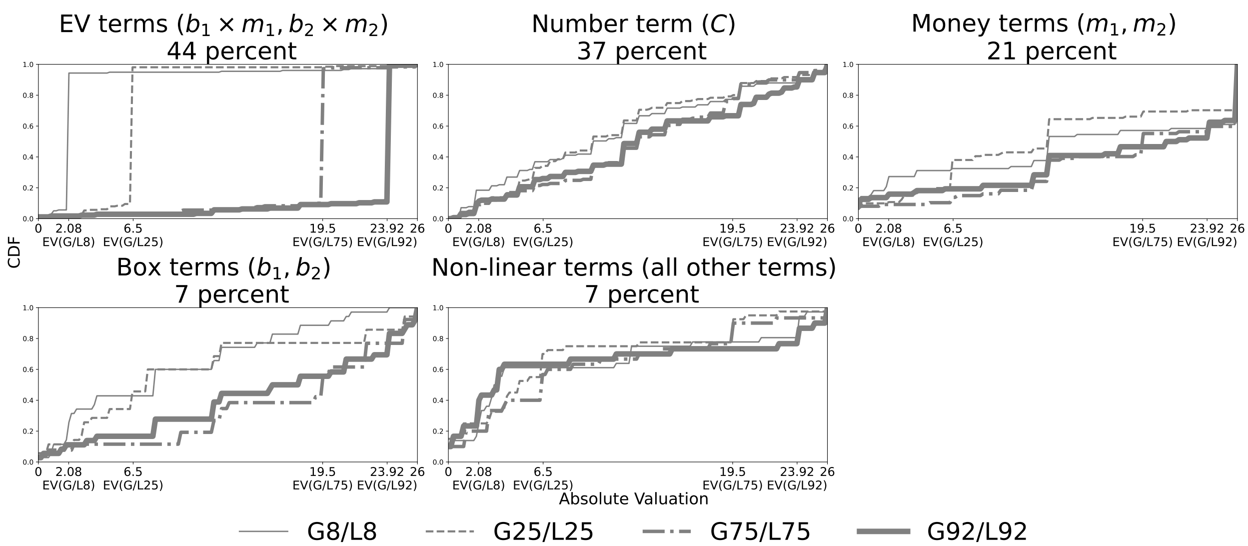


Figure A.2: CDF of mirror valuations in the NoCalc treatment, conditional on employing each group of base terms.

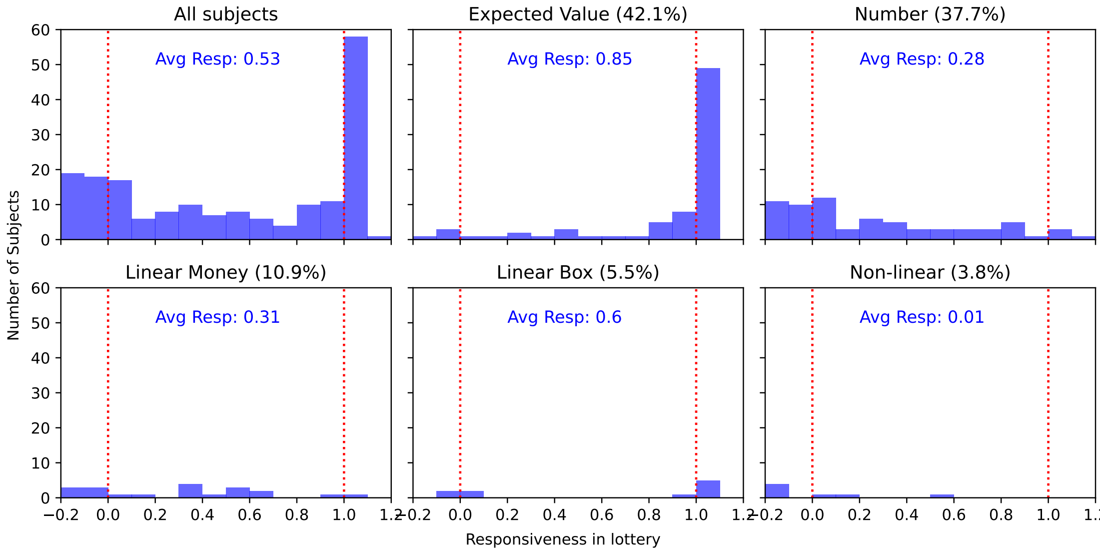


Figure A.3: Histograms of individual responsiveness in lottery tasks in the Calc treatment, excluding those subjects who give the same valuation in no fewer than 7 out of a total of 9 tasks, and separately for each type of subjects.

B Topic Modeling of Calculator Inputs

Latent Dirichlet Allocation (LDA) is an unsupervised machine learning technique designed to discover hidden thematic structures in collections of documents (Blei, Ng and Jordan, 2003). In the context of this paper, I apply LDA to uncover natural groupings in the calculator inputs provided by subjects.

LDA operates on the intuition that documents (in this case, calculator inputs) can be represented as mixtures of *topics*, where each topic is characterized by a distribution over words (in this case, base terms). LDA simultaneously estimates both the topic composition of each document and the word distribution of each topic. Formally, LDA models each topic $k \in \{1, 2, \ldots, K\}$ as a multinomial distribution $\boldsymbol{\phi}_k$ over the vocabulary of base terms. Each $\boldsymbol{\phi}_k$ is a vector where the component $\phi_{k,v}$ represents the probability of base term v appearing in topic k. These probabilities satisfy $\sum_v \phi_{k,v} = 1$ for all topics k. For each calculator input d, LDA also estimates a topic mixture $\boldsymbol{\theta}_d$, where each component $\theta_{d,k}$ represents the proportion of terms in document d that are drawn from topic k.

The key advantage of using LDA in this study is that it allows for the identification of "semantic" relationship between base terms without imposing a predetermined structure. Rather than manually categorizing base terms, LDA provides an unsupervised, data-driven approach to uncovering natural groupings based on how base terms co-occur within calculator inputs. This helps validate the intuitive categorization of base terms into the five groups (Expected value, Number, Linear money, Linear box, and Non-linear) used in the main analysis.

To implement the LDA model, I represented each calculator input in lottery tasks as a "document" defined by its corresponding base term set. This approach treats the collection of base terms associated with a calculator input as analogous to the words in a text document. For model specification, I set the number of topics (a hyperparameter that needs to be manually set) $K = 5$ to align with the five base term groups hypothesized in the main text. This parameter choice facilitates direct comparison between the data-driven topics and the conceptually defined groups.

Appendix Figure B.1 presents the LDA-generated topics and the associated probabilities of base terms appearing in each topic. The clustering of specific base terms within topics reflects their tendency to co-occur in subjects' calculations, providing a natural basis for grouping functionally similar terms. The results strongly support the classification of base terms into the five groups used in the main analysis. Topic 0 (34.1%) is dominated by terms involving the product of box quantities and monetary amounts ($b_1 \times m_1, b_2 \times m_2$), clearly corresponding to the expected value group. Topic 1 (33.4%) is primarily characterized by the constant term C , validating the number group. Topic 2 (19.5%) shows high probabilities for the monetary terms m_1 and m_2, aligning with the linear money group. Topic 3 (9.9%) features box quantities b_1 and b_2 as its most prominent terms, corresponding to the linear box group. Finally, Topic 4 (3.1%) captures various non-linear combinations of primitives ($b_2 \times m_1$, $1/b_1$, etc.) that do not fit into the other categories, supporting the non-linear group. This unsupervised classification thus provides strong empirical validation for the five base term groups employed throughout the paper.

It is worth noting that LDA estimation involves random initialization, which can affect the resulting topic and word distributions. Different initializations may produce somewhat different topic structures. However, in testing with multiple random initializations, I found that the vast majority of estimation runs generated at least 3-4 topics that clearly corresponded to the base term groups described above.

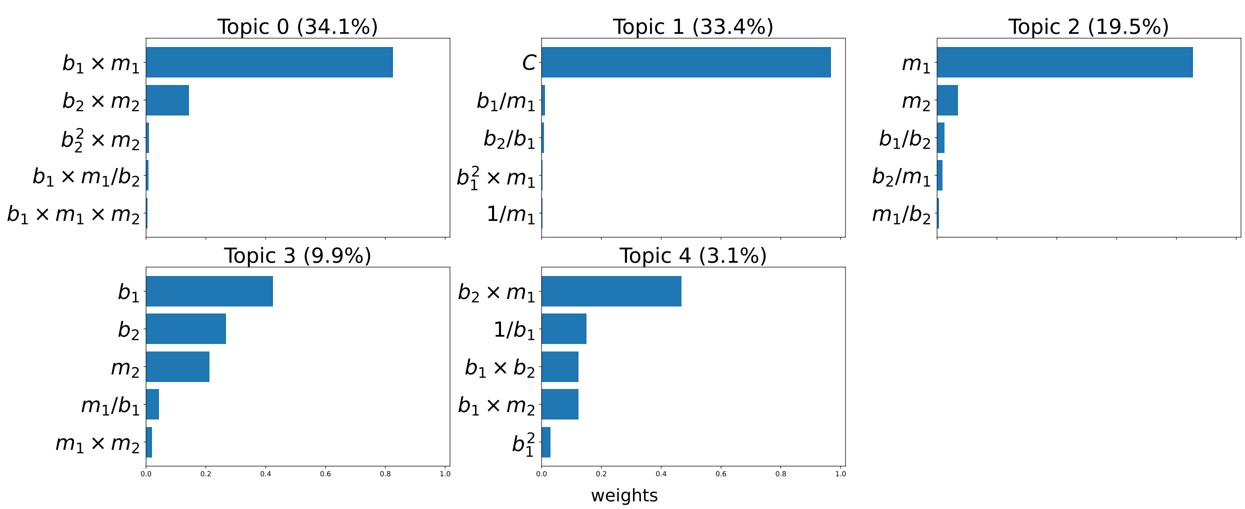


Figure B.1: LDA-generated topics and the probabilities of base terms in each topic

The topic model results can also be leveraged to construct an alternative approach to subject categorization. Rather than using modal base terms as described in Section 5.2, I aggregated the topic distributions of all calculator inputs to the subject level by averaging the document-topic mixtures ($\boldsymbol{\theta}_d$) across all documents (calculator inputs) produced by each subject in lottery tasks. This produces a subject-level topic distribution. I then classified each subject according to their dominant topic-the topic with the highest average probability. For example, subjects whose calculator inputs showed the highest average probability for Topic 0 (expected value) were classified as expected value type subjects. Remarkably, this LDA-based classification method yielded 95.0% agreement with the modal base term approach used in the main text. This high level of consistency between two methodologically distinct approaches to subject categorization provides strong evidence for the robustness of the subject type classifications and further validates the analysis of type-specific behaviors presented in the paper.

C Example Calculator Inputs

Below, I show randomly drawn example calculator inputs corresponding to the top 15 most frequent base term sets in lottery rounds. For each base term set, I give two examples, first a lottery round and then a mirror round. These top 15 base term sets collectively account for 90.5% of lottery rounds and 88.1% of mirror rounds.

Please note that, the experimental program elicits the willingness to *pay* for the lottery/mirror in gain tasks. However, the program instead elicits willingness to *prevent* (the maximum amount willing to forgo to prevent the lottery/mirror from happening) in loss tasks. In these tasks, the certainty equivalent is the negative of the willingness to prevent. As a result, the last row of the *Result* column represents the valuation in gains tasks, but instead the negative of the valuation in loss tasks. See Appendix G for the complete instructions.

- Base term set: $\{C\}$, frequency in lottery: 28.82%, frequency in mirror: 30.58%
  - Subject ID: 5iu9eyzk
  - Task: L25 lottery

Line	Numerical Expression	Result
1	24	24

  - Subject ID: 6n65prfw
  - Task: L92 mirror

Line	Numerical Expression	Result
1	7×3	21

- Base term set: $\{b_1 \times m_1\}$, frequency in lottery: 27.39%, frequency in mirror: 29.92%
  - Subject ID: 88171s3p
  - Task: L8 lottery

Line	Numerical Expression	Result
1	$8/100 \times 26$	2.08

- Subject ID: 7vzs5is7
- Task: L25 mirror

Line	Numerical Expression	Result
1	26×25	650
2	$650/100$	6.5

- Base term set: $\{m_1\}$, frequency in lottery: 13.20%, frequency in mirror: 10.29%
  - Subject ID: hgqqa7bl
  - Task: L75 lottery

Line	Numerical Expression	Result
1	26	26

  - Subject ID: id6e5867
  - Task: L92 mirror

Line	Numerical Expression	Result
1	26×1	26

- Base term set: $\{b_1 \times m_1, b_2 \times m_2\}$, frequency in lottery: 4.95%, frequency in mirror: 5.50%
  - Subject ID: cef3c6ir
  - Task: L25 lottery

Line	Numerical Expression	Result
1	$25 \times 26 + 75 \times 0/100$	650
2	$650/100$	6.5

- Subject ID: cef3c6ir
- Task: G92 mirror

Line	Numerical Expression	Result
1	$92 \times 26 + 8 \times 0$	2392
2	$2392/100$	23.92

- Base term set: $\{C, b_1 \times m_1\}$, frequency in lottery: 3.03%, frequency in mirror: 2.31%
  - Subject ID: qqoqux4t
  - Task: G25 lottery

Line	Numerical Expression	Result
1	$26 \times .25$	6.5
2	13	13

  - Subject ID: sysy515p
  - Task: G25 mirror

Line	Numerical Expression	Result
1	25×26	650
2	$Ans1/100$	6.5
3	$Ans2 - 3$	3.5

  Please note that the calculator has an Ans button, which is a shortcut to use the result from the previous calculation.

- Base term set: $\{C, m_1\}$, frequency in lottery: 2.75%, frequency in mirror: 1.60%
  - Subject ID: wshvt3vq
  - Task: G75 lottery

Line	Numerical Expression	Result
1	$26/2 + 1$	14

- Subject ID: 668q3qjp
- Task: G25 mirror

Line	Numerical Expression	Result
1	26/2	13
2	$Ans1 - 8$	5

- Base term set: $\{m_2\}$, frequency in lottery: 1.98%, frequency in mirror: 1.21%
  - Subject ID: lhxiwlwe
  - Task: L8 lottery

Line	Numerical Expression	Result
1	2×0	0

  - Subject ID: l5hdvvcc
  - Task: L8 mirror

Line	Numerical Expression	Result
1	0	0

- Base term set: $\{b_1\}$, frequency in lottery: 1.38%, frequency in mirror: 1.82%
  - Subject ID: 9ny75gkz
  - Task: G25 lottery

Line	Numerical Expression	Result
1	30×0.25	7.5

  - Subject ID: y1vfs7ou
  - Task: G25 mirror

Line	Numerical Expression	Result
1	$30 \times .25$	7.5

- Base term set: $\{m_1, m_2\}$, frequency in lottery: 1.32%, frequency in mirror: 1.32%
  - Subject ID: 7k9gkzry
  - Task: L25 lottery

Line	Numerical Expression	Result
1	0.	0
2	26.00	26

  - Subject ID: pgwtf1kz
  - Task: G75 mirror

Line	Numerical Expression	Result
1	$26 + 0$	26

- Base term set: $\{b_1, b_1 \times m_1, b_2, b_2 \times m_2\}$, frequency in lottery: 1.27%, frequency in mirror: 0.00%
  - Subject ID: si04ncfg
  - Task: L25 lottery

Line	Numerical Expression	Result
1	25/100	0.25
2	75/100	0.75
3	$0.25 \times (26) + 0.75 \times 0$	6.5

- Base term set: $\{b_1, b_1 \times m_1\}$, frequency in lottery: 1.16%, frequency in mirror: 0.94%
  - Subject ID: 7bfat155
  - Task: G25 lottery

Line	Numerical Expression	Result
1	25/100	0.25
2	0.25×26	6.5

- Subject ID: n4kcm1vv
- Task: G75 mirror

Line	Numerical Expression	Result
1	75×261	19575
2	75×26	1950
3	1950/100	19.5

- Base term set: $\{b_2 \times m_1\}$, frequency in lottery: 0.88%, frequency in mirror: 0.44%
  - Subject ID: 98qa7bdb
  - Task: G92 lottery

Line	Numerical Expression	Result
1	$.08 \times 26$	2.08

  - Subject ID: ze63xdkx
  - Task: L75 mirror

Line	Numerical Expression	Result
1	26×0.25	6.5

- Base term set: $\{C, m_2\}$, frequency in lottery: 0.83%, frequency in mirror: 0.33%
  - Subject ID: 00f2schi
  - Task: G25 lottery

Line	Numerical Expression	Result
1	$5 + 0$	5

  - Subject ID: ehfz4yq0

- Task: L92 mirror

Line	Numerical Expression	Result
1	$0 + 6$	6

- Base term set: $\{b_1, b_2\}$, frequency in lottery: 0.77%, frequency in mirror: 0.94%
  - Subject ID: dwiup06z
  - Task: G25 lottery

Line	Numerical Expression	Result
1	$75 + 25$	100
2	$100/2$	50
3	$50/4$	12.5

  - Subject ID: dwiup06z
  - Task: L8 mirror

Line	Numerical Expression	Result
1	$92 + 8$	100
2	$100/2$	50
3	$50/4$	12.5

- Base term set: $\{b_2\}$, frequency in lottery: 0.77%, frequency in mirror: 0.88%
  - Subject ID: i1uphqli
  - Task: L25 lottery

Line	Numerical Expression	Result
1	$75/100$	0.75
2	0.75×30	22.5

  - Subject ID: i1uphqli

– Task: G25 mirror

Line	Numerical Expression	Result
1	75/100	0.75
2	0.75×30	22.5

D Robustness of the Similarity of Lottery and Mirror Valuations

Comprehension questions and errors The comprehension questions in my experiment are taken from Oprea (2024). However, I only include the first three out of the four comprehension questions in Oprea (2024) (see Appendix G). Before each of the four blocks (see Figure 2), I ask each subject the same set of three questions. If they answer any of the questions incorrectly, the incorrect answers will be cleared from their interface. They are informed of their error and must answer again. Subjects are given unlimited opportunities and can not continue until they get the correct answer.

I record a subject-level variable `errors` that is set to 0 when the experiment starts. Each time a subject incorrectly answers any set of comprehension questions, I add one to the errors variable of that subject. In other words, the variable errors records the total number of trials the subject takes before they complete all four sets of comprehension questions. This variable is the only thing I record related to the errors subjects make when responding to the comprehension questions. Particularly, I do not record which specific question or which specific block was answered incorrectly.

Out of my 202 subjects, there are 49 subjects with errors = 0, which means that they pass all four sets of comprehension questions at the first try. Since these subjects are equivalent to those subjects who are defined as passed comprehension check by Banki, Simonsohn, Walatka, and Wu (2025, BSWW), I refer to these subjects as BSWW subjects.

Replication of the Patterns in BSWW First, aggregating across all subjects, my data generate similar lottery and mirror valuations. I compared the distribution of lottery valuations with mirror valuations using Kolmogorov-Smirnov (K-S) tests for all 16 combinations of treatment (2 in total) $\times$ task (8 in total). None of these comparisons yielded p-values below 0.05. Only one comparison had a p-value below 0.1, and three had p-values below 0.2.

Appendix Figure D.1 shows the median valuations for all lotteries and mirrors in both NoCalc and Calc treatments, pooling all subjects. The median valuations are similar for lotteries and mirrors, in all the four panels. For the NoCalc treatment, for both lottery and mirror tasks, the fourfold patterns are visible in this graph when measured by the median. On the opposite, for the Calc treatment, for both lottery and mirror tasks, the fourfold patterns are almost non-existent when measured by the median[26]. But the most important takeaway is that, fourfold patterns in lotteries and mirrors appear and disappear together.

Second, the similarity of lottery and mirror valuations are robust if I only consider those 49 BSWW subjects. When conducting the same set of 16 K-S tests above, I again cannot reject any of these null hypotheses after applying standard multiple hypothesis testing corrections. Appendix Figure D.2 shows the median valuations for all lotteries and mirrors in both NoCalc and Calc treatments, only including the BSWW subjects. Again, fourfold patterns in lotteries and mirrors appear and disappear together even in this subset of subjects. Especially, opposite from the findings using the original data from Oprea (2024), the BSWW subjects did not exhibit meaningful fourfold patterns in the lottery tasks.

Moreover, I construct the π statistic (defined by BSWW's Equation 1), and Appendix Figure D.3 plots the average of π by quartile of the errors variable. This exercise aims to replicate the upper-left panel of Figure 9 in BSWW. Again, opposite to the findings using the data from Oprea (2024), I don't find π to systematically change with quartile of errors.

Finally, echoing the findings in BSWW, my subjects exhibit a similar degree of first-order stochastic dominance (FOSD) violations as Oprea (2024), which is much higher than in previous papers. Since this study adopts heavily from Oprea (2024)'s instructions and use the same Prolific subject pool, the similar degree of FOSD violations is not surprising. Specifically, for both lottery and mirror tasks, around 20% of subjects exhibit the FOSD violation of valuing G8/L8 higher (in absolute terms) than G92/L92. The numbers by quartile of errors are quantitatively similar with Figure 5 in BSWW.

[26]From my recorded calculator input data, more than 40% of subjects calculated the expected value for both lotteries and mirrors, and the vast majority of them actually submitted the expected value as their valuation. Using the median (as opposed to the mean) to summarize the subjects' valuations hides the valuation patterns of most other subjects.

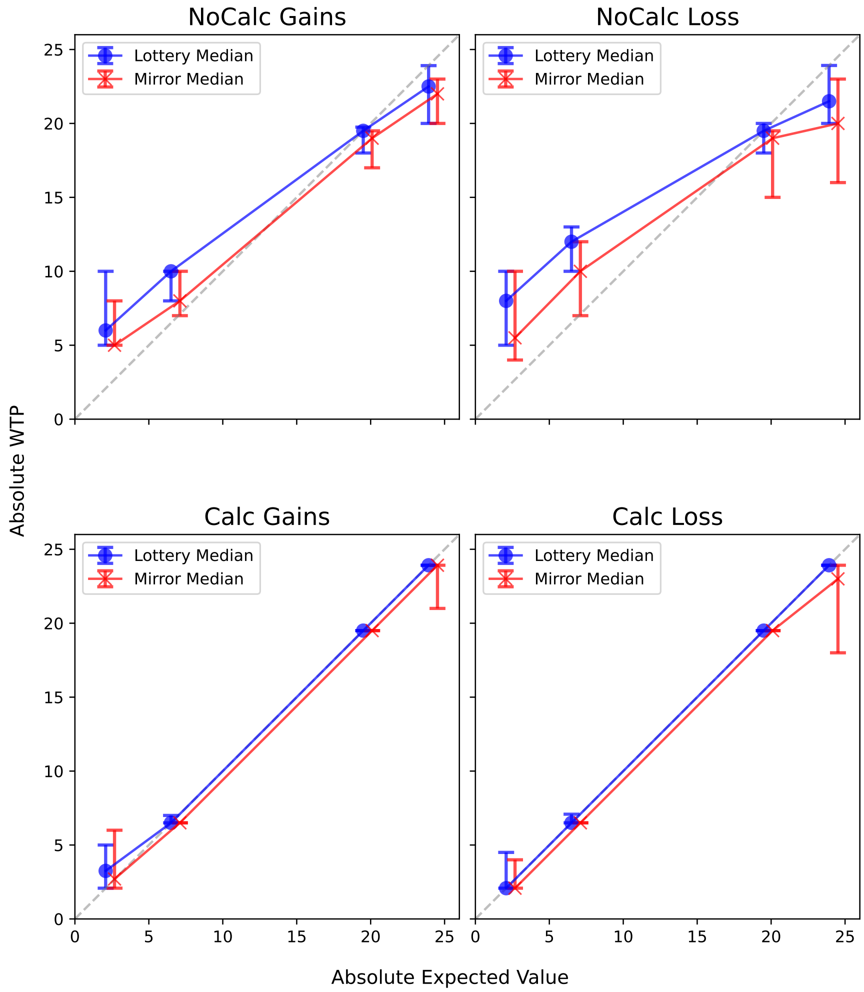


Figure D.1: Median valuations for each combination of $\{gains, losses\} \times \{lottery, mirror\}$, pooling all subjects. The 95% confidence interval is obtained by bootstrap. A small offset is added to make the confidence intervals easier to see.

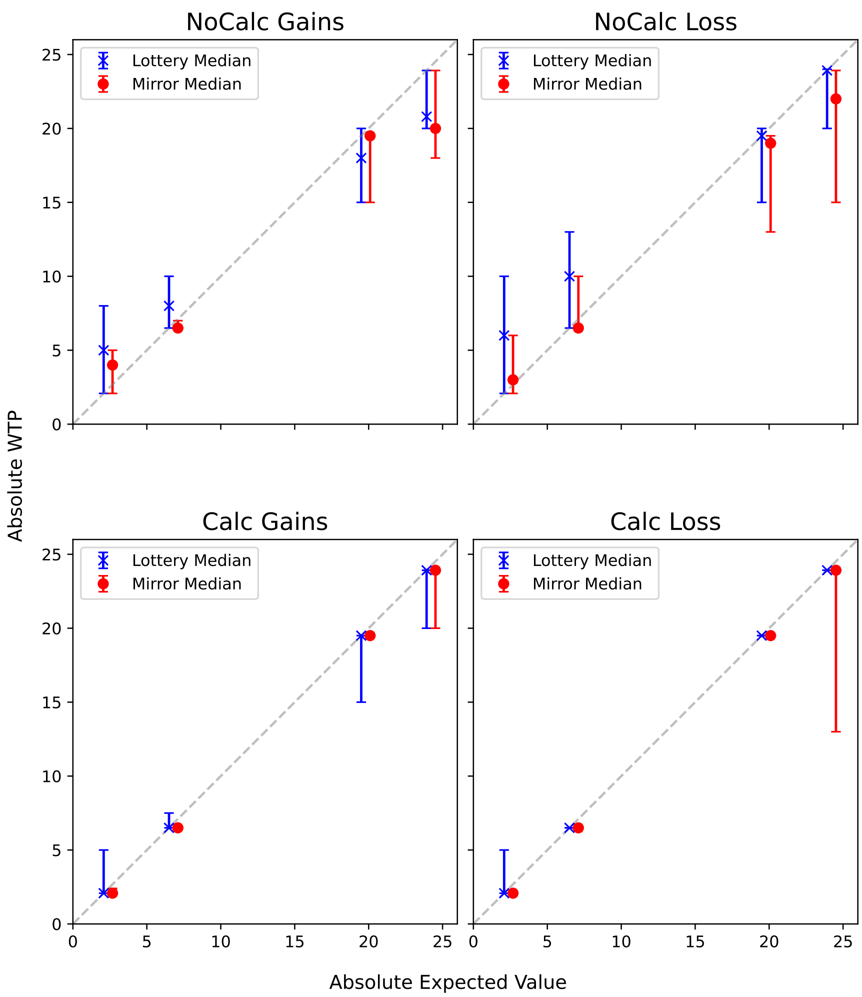


Figure D.2: Median valuations for each combination of $\{gains, losses\} \times \{lottery, mirror\}$, for all BSWW subjects. The 95% confidence interval is obtained by bootstrap. A small offset is added to make the confidence intervals easier to see.

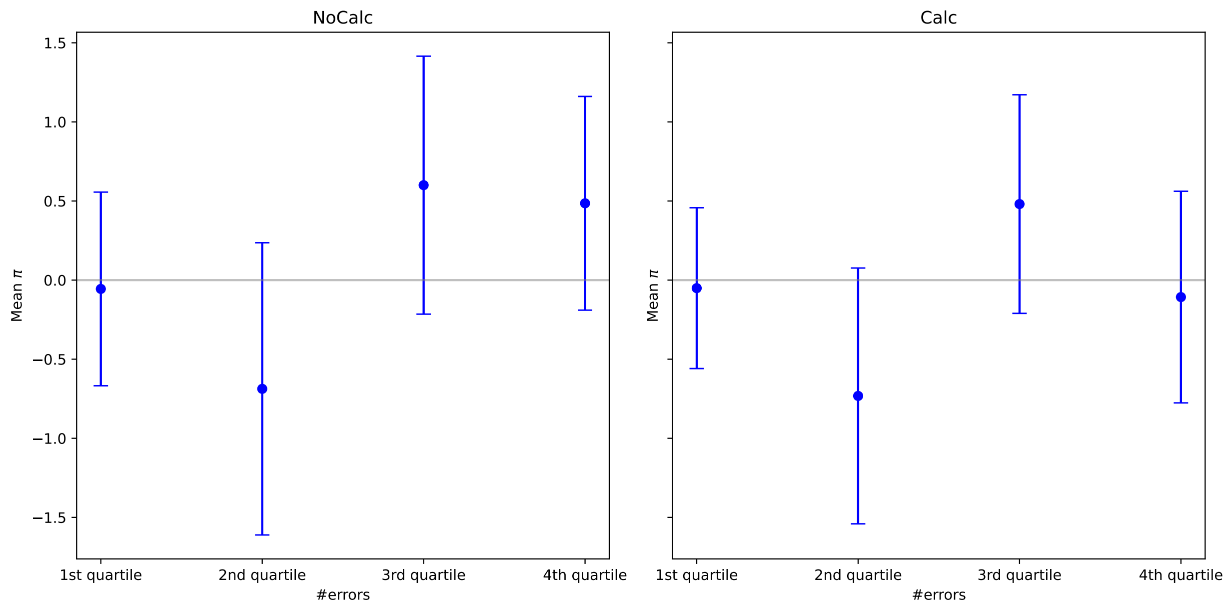


Figure D.3: Average π by quartile of errors

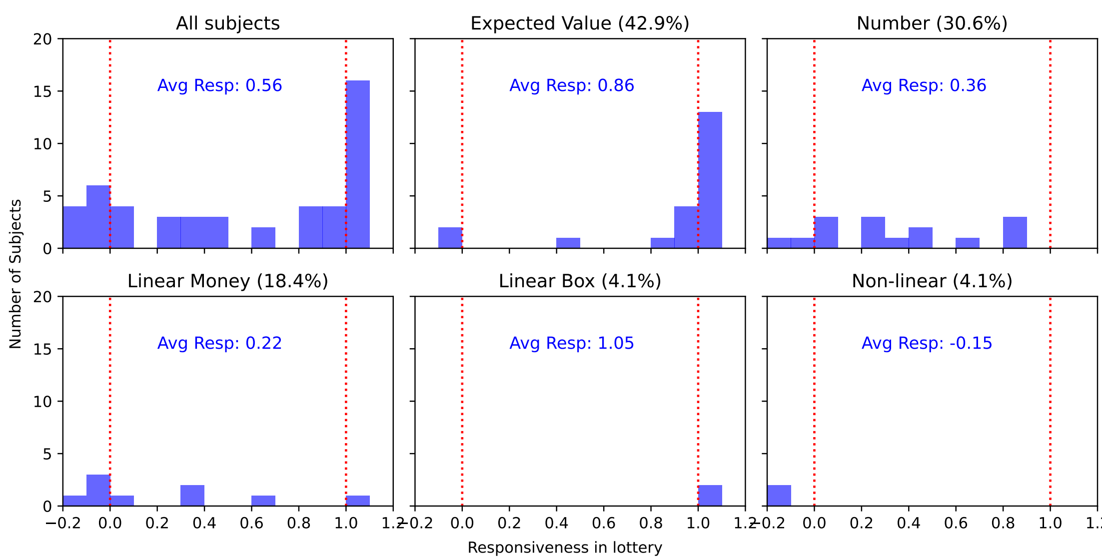


Figure D.4: Histograms of individual responsiveness in lottery tasks in the Calc treatment, only including the 49 subjects who answer all four batches of comprehension questions correctly at their first trial.

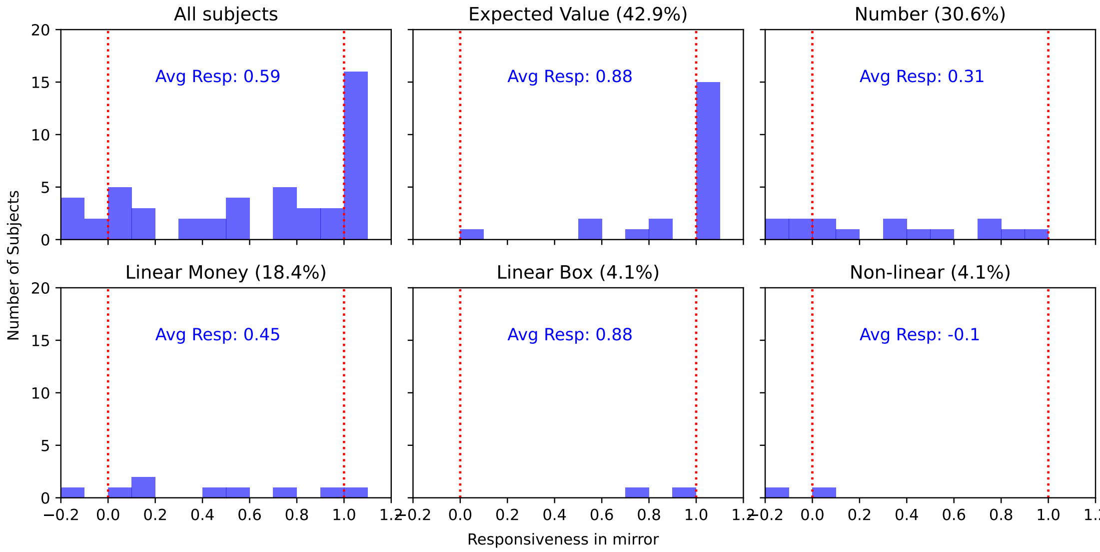


Figure D.5: Histograms of individual responsiveness in mirror tasks in the Calc treatment, only including the 49 subjects who answer all four batches of comprehension questions correctly at their first trial.

E Reconciliation and Survey

E.1 Reconciliation

type	Calc	NoCalc	Keep Inconsistent
expected value	55%	17%	28%
number	38%	29%	33%
linear money	44%	20%	35%
linear box	27%	30%	43%
nonlinear	57%	19%	24%

Table E.1: Caption

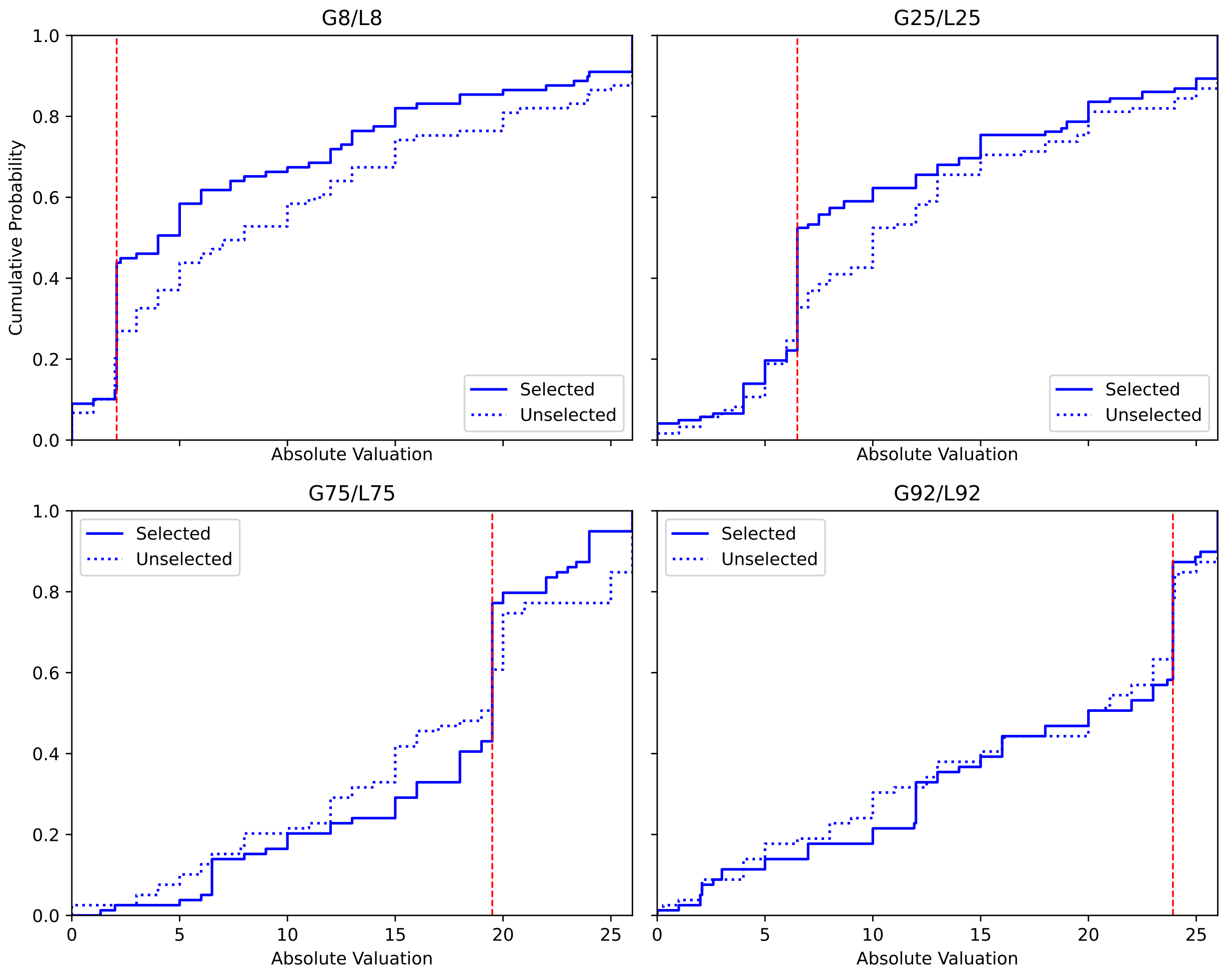


Figure E.1: Caption

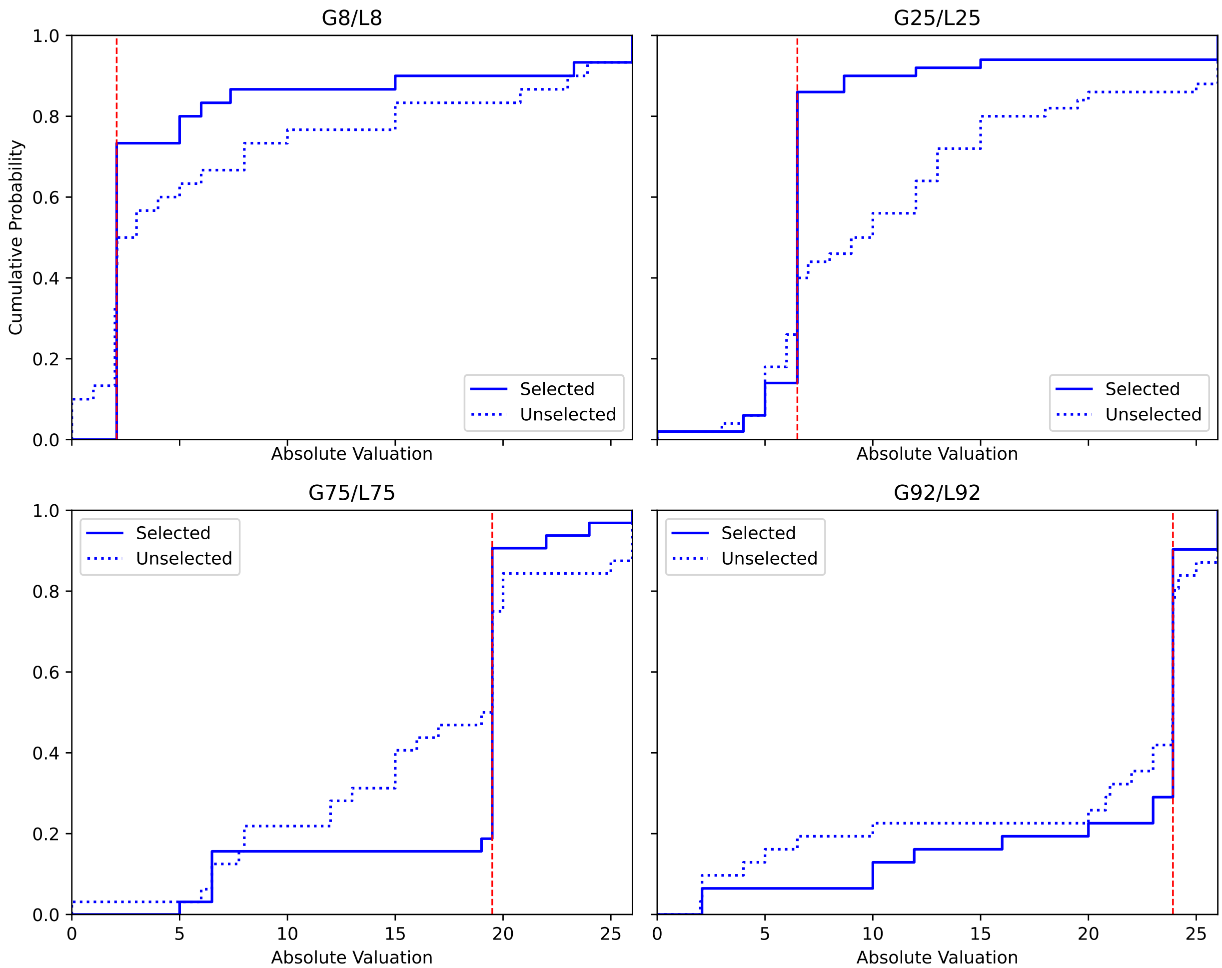


Figure E.2: Caption

E.2 Survey

type	%(ev=6)	%(dominant chosen)
expected value	96.1%	93.5%
linear box	60.0%	90.0%
linear money	58.8%	88.2%
nonlinear	42.9%	100.0%
number	71.6%	81.1%

Table E.2: Caption

type	%(ev=6)	resp|ev=6	resp|ev $\neq$ 6
expected value	0.96	0.85	0.86
linear box	0.60	0.84	0.24
linear money	0.59	0.28	0.08
nonlinear	0.43	-0.20	0.04
number	0.72	0.30	0.13

Table E.3: Caption

F Mathematical Expressions and Algorithms

This section first introduces a tree structure of mathematical expressions. Then, it explains the algorithm with which I recover the symbolic expressions, and the algorithm constructing the base terms. I implement the tree structure and all the algorithm using the open-source Python package `sympy`.

F.1 Mathematical Expressions as Trees

A mathematical expression can be represented as an *expression tree*. The tree has a few features:

- Leaf nodes represent operands (numbers or symbols).
- Non-leaf nodes represent operators (such as $+$, $\times$, or Power).
- Each non-leaf node generates a subtree, representing a sub-expression.
- Both $+$ and $\times$ are defined as n-ary operators (as opposed to binary operators), meaning they can take any finite number of arguments.
- The operators $-$ and $/$ are represented in the tree using $+$, $\times$, and Power. For example, $a - b$ is represented as $a + (-1 \times b)$, and a/b is represented as $a \times \text{Power}(b, -1)$.
- The parent-child structure of the tree represents the order of operators. A parent operator is evaluated later than its children.

As an example, Appendix Figure F.1 shows the tree representation of the expression $5 \times 4 + 3 - 1$. This representation preserves the order of operations because $+$ is a parent node of the $\times$. For more information, I refer the Reader to the sympy documentation.

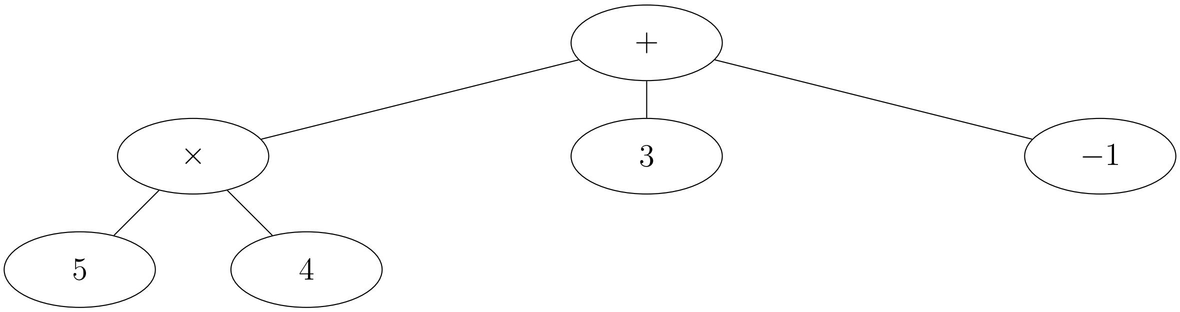


Figure F.1: The expression tree representing $5 \times 4 + 3 - 1$

For my purpose, one of the most important features of sympy's implementation of this expression tree is that any subtree with root operator $+$ or $\times$ will automatically "flatten" itself to the shallowest subtree possible. For example, an alternative (and illegal) expression tree representing the expression $5 \times 4 + 3 - 1$ is shown in Appendix Figure F.2. Since the operator $+$ is allowed to be n-ary (as opposed to binary), the right subtree will be automatically flattened to form the shallower tree in Appendix Figure F.1. For more information, see here for $+$ and here for $\times$.

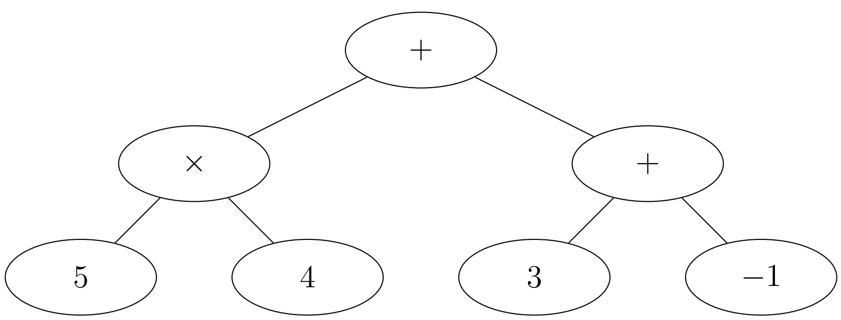


Figure F.2: An unflattened expression tree representing $5 \times 4 + 3 - 1$

F.2 Recovering Symbolic Expressions

I denote each calculator input as its sequence of calculator lines $L = (l_n)_n$, where l_n represents the numerical expression (represented as an expression tree) in line $n \in \{1, \ldots, \bar{n}\}$. I also use r_n to denote the numerical result from evaluating the numerical expression l_n. Let $P = \{(x_n, v_n)\}_n$ denote the set of task primitives that the algorithm matches against, where x_n represents a symbol or a symbolic sub-expression, and v_n represents its corresponding numerical value.

When implementing the algorithm, the following symbolic primitives are being matched:

$$\{b_1, b_2, m_1, m_2, b_1/100, b_2/100, m_1/100, b_1 \times m_1, b_1 \times m_1/100\}.$$

The set of primitives is expanded beyond $\{b_1, b_2, m_1, m_2\}$ to include common calculation shortcuts that subjects may use. For example, in G75, the primitive set includes $(b_1/100, 0.75)$ to capture the possibility that a subject divides the number of boxes by 100 implicitly in their mind before they use the result of this mental calculation directly in the calculator.

Having defined the necessary notations, I now describe the matching algorithm that recovers the symbolic expression from numerical expressions. Starting with the first line l_1, the algorithm iterates through its all leaf nodes (the numbers in the expression). If a number in a leaf node matches some v_n, the node's content is replaced with the corresponding symbolic sub-expression x_n. Any leaf node with a number that does not match any primitive remains as the same node. In this way, the algorithm constructs the symbolic expression for the first line, s_1.

Since the calculations made in a line $n > 1$ may build upon previous results, the algorithm must iteratively incorporate these intermediate calculations. Therefore, for each subsequent line $n > 1$, the algorithm expands its matching set to $M_n = (\cup_{i<n}(s_i, r_i)) \cup P$, which includes both the original primitives and all previous line results. Specifically, $\cup_{i<n}(s_i, r_i)$ contains the pairs of symbolic expressions and their computed results from all previous lines $i < n$. When processing line n, for any leaf node with a number that matches a previous result r_i, its content is replaced with the corresponding symbolic expression s_i. In the meantime, matches with primitive values v_k continue to be replaced with x_k. This process yields the symbolic expression s_n for each line n.

Through this iterative construction, the algorithm generates a sequence of symbolic expressions $S = (s_n)_n$ from the numerical expressions L.

F.3 Terms and Base Terms

For any symbolic expression l, if its root node is the operator $+$, its *terms* are all the second-level subtrees (whose roots are the immediate child nodes of the root node) of the symbolic expression. Otherwise, the symbolic expression has only one term: the symbolic expression itself. The flattening property of the expression trees ensures that all base terms have roots (the upper-level operand) other than the operator $+$. For example, $b_1 + b_2$ cannot be a term, while b_1 and $b_1 \times m_1$ are permitted. I implement the algorithm backing out terms via the function `as_ordered_terms` (see here) in sympy.

Then, for each term t, I first find all its factors. If the root node of a term is $\times$, the factors are all the second-level subtrees. Otherwise, the only factor of this term is the term itself. See `as_ordered_factors` documentation for more information. Finally, I drop all factors which are a single number (as opposed to symbols, or sub-expressions) from the term to generate its corresponding *base term*. If all factors are number factors (as opposed to symbols, or sub-expressions), the base term is defined as C.

The concept of terms, and by extension base terms, runs into problems with syntactically different but mathematically equivalent expressions – for example, the mathematically equivalent expressions $b_1 \times m_1/100 + b_2 \times m_2/100$ and $(b_1 \times m_1 + b_2 \times m_2)/100$ lead to different terms and in turn base terms. To address this problem, I first expand all the products in all expressions by applying the distributive law of multiplication ($a \times (b + c) = a \times b + a \times c$), wherever applicable. This way, I transform the original symbolic expression into its distributed form expression. Using distributed form expressions solves the aforementioned indeterminacy problem and generates the same set of base terms ($\{b_1 \times m_1, b_2 \times m_2\}$) for $b_1 \times m_1/100 + b_2 \times m_2/100$ and $(b_1 \times m_1 + b_2 \times m_2)/100$.

G Experimental Instructions

General Information

Welcome to our study on decision-making.

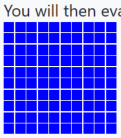
Participation in this study guarantees a $7.00 show-up fee. Additionally, you have the opportunity to earn a bonus. How much bonus you earn will depend partly on the decisions you make and partly on chance.

This session includes three Parts. You will be asked to make decisions in each Part, and the first two Parts offer you an opportunity to earn a bonus. The total bonus paid to you at the end of this session will be the sum of the bonuses you earn in all Parts.

Next

Boxes with Money

- In each of several tasks, we will give you an **INITIAL** sum of money of **$30**.
- You will then evaluate a set of **100 BOXES** which the computer may open to **either increase or decrease** this initial sum.



- Each box contains either a **POSITIVE** or **NEGATIVE** amount of money (or nothing). When the computer opens one or more boxes from a set, the amount of money in the opened boxes will be added to (or subtracted from) your INITIAL money to determine your **BONUS**.

Next

The Decision Table

- Each set of boxes will be described in a **TABLE** like the one below. For each set, one or more counts of boxes (for instance 75, 25 or 100 boxes) are listed at the top, and the positive or negative amount of money in that number of boxes (for instance $20, $0, $7) is shown in the row of the Table.
-
75 Boxes	25 Boxes
$20.00	$0.00

  In the example above, the set consists of 75 boxes each containing $20 and 25 boxes each containing $0.
- In the example below, the set consists of 25 boxes with -$12 (negative $12) in each box and 75 boxes with $0 in each box.

25 Boxes	75 Boxes
- $12.00	$0.00

- Depending on the task, your job will be to decide how much you'd be willing to pay to either **cause** the computer to open boxes from the set to modify your **BONUS** or **prevent** the computer from opening the boxes to modify your bonus.

Next

If the next tasks are lottery tasks, the following instructions appear.

A Random Box

- In the upcoming tasks, if the computer opens boxes, it will pay you by **RANDOMLY** selecting one of the 100 boxes (each box in the set is **EQUALLY** likely to be selected by the computer). If the amount in the box is positive, it will be **ADDED** to your initial money. If the amount is negative, it will be **SUBTRACTED** from your initial money.

- Example: In the example below, there are 100 boxes in the set. For this set, 50 boxes contain $16.00 and 50 of them contain $0.00. If the computer opens the boxes, there is therefore a 50% chance that $16 will be added to your initial amount of money and a 50% chance $0 will be added.

50 Boxes	50 Boxes
$16.00	$0.00

- Example: In the example below, there are also 100 boxes in the set. For this set, 50 boxes contain -$8.00 and 50 of them contain $0.00. If the computer opens the boxes, there is therefore a 50% chance you will have $8 subtracted from your initial amount of money (you lose $8) and a 50% chance that you have $0 subtracted.

50 Boxes	50 Boxes
- $8.00	$0.00

- Depending on the task, your job will be to decide how much you'd be willing to pay to either cause the computer to open boxes from the set to modify your **BONUS** or prevent the computer from opening the boxes to modify your bonus.

3 comprehension questions are asked on the same page as the instructions over the payoff rules of lotteries. The subject needs to simultaneously answer all 3 questions correctly to proceed.

Please answer the questions below.

50 Boxes	50 Boxes
$16.00	$0.00

Suppose that this example of boxes determines your payment in the upcoming tasks.

For the upcoming tasks, how will your payment be decided by the boxes?

- ○ The maximum box
- ○ The minimum box
- ○ The average box
- ○ A random box

What is the chance that $16 is added to your earnings?

- ○ 0 in 100 (0%)
- ○ 50 in 100 (50%)
- ○ 100 in 100 (100%)

What is the chance that $8 is added to your earnings?

- ○ 0 in 100 (0%)
- ○ 50 in 100 (50%)
- ○ 100 in 100 (100%)

What is the chance that $4 is added to your earnings?

- ○ 0 in 100 (0%)
- ○ 50 in 100 (50%)
- ○ 100 in 100 (100%)

Next

Otherwise if the next tasks are mirror tasks, the following instructions appear, and the same 3 comprehension questions are asked.

The Average Box

- In the upcoming tasks, if the computer opens boxes, it will pay you by calculating the **AVERAGE** amount of money across all 100 boxes. That is, it will add up the amount of money from each of the 100 boxes and divide that sum by 100. If the amount is positive, that amount will be **ADDED** to your initial money. If the amount is negative, it will be **SUBTRACTED** from your initial money.

- Example: In the example below, there are 100 boxes in the set. For this set, 50 boxes contain $16.00 and 50 of them contain $0.00. If the computer opens these boxes, it will therefore add $8 to your initial amount of money for certain, which is the **AVERAGE** amount of money across the 100 boxes.

50 Boxes	50 Boxes
$16.00	$0.00

- Example: In the example below, there are also 100 boxes in the set. In this set, 50 boxes contain -$8.00 and 50 of them contain $0.00. If the computer opens these boxes, the computer will pay you -$4 for your choice, which is the **AVERAGE** amount of money across the 100 boxes; it will therefore subtract $4 from your initial amount of money (that is, you will lose $4) for certain.

50 Boxes	50 Boxes
- $8.00	$0.00

After introducing the payout rules of lottery (or mirror), instructions for the BDM mechanism is shown

Paying for a Set of Boxes

- In the experiment, we will ask you the maximum amount you would be willing to pay either to cause or prevent the computer from opening the set of boxes on your screen to modify your Initial earnings.

- In some tasks (colored in green) we will show you a set that contains positive amounts of money, and ask you to tell us how many dollars you would (**at the very maximum**) be willing to pay to **cause** the computer to open boxes from the set to increase your earnings. Or, in other words, we will ask you how much do you think it is worth to you to have these boxes influence your earnings?

- Example: On your screen, we will show you a text box like the one below. Just enter the amount of money you think the set is worth to you (the maximum amount you'd be willing to pay for the set to be opened - the screen will give you the range you can enter):

  I would be willing to pay **a maximum of**:

  (enter a number between $0 and $25.00)

- To **reward you** for giving an **honest answer**, we are going to use a special set of rules to determine your payments in these tasks. We will randomly pick a **price** (equally likely between 0 and the maximum value you are allowed to enter) for the set of boxes (you won't know the price when you make your choice). If the amount you entered is **greater than or equal to** that random price, the computer will open the set of boxes on the screen to modify your Initial earnings as described **and** you will pay the amount of the random price (not the amount you entered) from your total earnings. If your maximum amount is less than the random price, the computer will not open the boxes on your screen and you will simply earn your initial amount (and you will not pay the random price).

- Important: If the computer uses boxes from the set to modify your earnings, you will not have to pay the maximum amount you enter, but instead will pay the random price. The maximum amount you enter just lets you tell us the range of random prices you are willing to pay for the set of boxes.

- If this sounds confusing, it is **actually very simple**. We've designed the payments so it is in your best interest to **tell us honestly** the **most** you would be willing to pay to have the set of boxes opened to influence your bonus. So just think about how much at a maximum you'd be willing to give up to have the computer modify your bonus based on the set of boxes on your screen, and enter this amount truthfully.

- Example: In the example below, the set consists of $12 in each of the 100 boxes. If the computer opens the boxes, with certainty $12 will be added to your initial amount of money.

100 Boxes	0 Boxes
$12.00	$0.00

To maximize the bonus you get from this session, you should submit $12 as the maximum amount that you are willing to pay to have the computer open the boxes. Our payment rule guarantees this.

Paying to Avoid a Set of Boxes

- In other tasks (colored in red) we will show you a set that contains negative amounts of money, and ask you to tell us how many dollars you would (at the very maximum) be willing to pay to prevent the computer from opening boxes from the set to decrease your earnings. Or, in other words, we will ask you how much do you think it is worth to you to prevent these boxes from influencing your earnings?

- Example: On your screen, we will, again, show you a text box like the one below. Just enter the amount of money you think it is worth to prevent the computer from using that set to modify your bonus (the maximum amount you'd be willing to pay to prevent it - the screen will give you the range you can enter):

  I would be willing to pay **a maximum of**:

  (enter a number between $0 and $25.00)

- To **reward you** for giving an **honest answer**, we are going to use a special set of rules to determine your payments in these tasks. We will randomly pick a **price** (equally likely between 0 and the maximum value you are allowed to enter) for the set of boxes (you won't know the price when you make your choice). If the amount you entered is **greater than or equal to** that random price, the computer will **not open** the set of boxes on the screen to modify your Initial earnings as described **and** you will pay the amount of the random price (not the amount you entered) from your total earnings. If your maximum amount is less than the random price, the computer will open the boxes on your screen and you will simply earn your initial amount (and you will not pay the random price).

- Important: If you prevent the computer from opening boxes from the set, you will not have to pay the maximum amount you enter, but instead will pay the random price. The maximum amount you enter just lets you tell us the range of random prices you are willing to pay to avoid the set of boxes.

- If this sounds confusing, it is **actually very simple**. We've designed the payments so it is in your best interest to **tell us honestly** the **most** you would be willing to pay to prevent the set of boxes from being opened to influence your bonus. So just think about how much at a maximum you'd be willing to give up to prevent the computer from modifying your bonus based on the set of boxes on your screen, and enter this amount truthfully.

- Example: In the example below, the set consists of -$12 in each of the 100 boxes. If the computer opens the boxes, with certainty $12 will be subtracted from your initial amount of money.

100 Boxes	0 Boxes
- $12.00	$0.00

To maximize the bonus you get from this session, you should submit $12 as the maximum amount that you are willing to pay to prevent the computer from opening the boxes. Our payment rule guarantees this.

Several Sets of Boxes

- Over the course of the session, we will show you several sets of boxes. Each gives you $30 **initial amount of money**, but may have **different** amounts of money distributed across the boxes.

- Important: Make sure you pay attention to the type of question we are asking in each task. In some tasks colored in green we are asking you to tell us how much you'd be willing to pay to **cause** the boxes to influence your earnings. In other tasks colored in red we are asking you to tell us how much you'd be willing to pay to **prevent** the boxes from influencing your earnings.

- **One out of five (1/5 of)** participants will be randomly selected by the computer to be paid a **BONUS** based on their choices. If you are one of these participants, at the end of the session the computer will **RANDOMLY** select **ONE** Task and then **RANDOMLY** select a **PRICE** to determine your payment based on how much you said you're willing to pay.

- Since you do not know which choice will be selected, you should make each choice as if it alone determines your payment.

Experimental interface in the NoCalc treatment:

Initial money: $30.00

75 Boxes	25 Boxes
$26.00	$0.00

I would be willing to pay **a maximum of**:

(enter a number between $0 and $26.00)

to **have** a **randomly selected** box's contents **added to** my Initial Money.

Remember, we've designed the payments so it is in your best interest to **tell us honestly** the most you would be willing to pay to have the set of boxes opened to influence your bonus. So just think about **how much at a maximum you'd be willing to give up** to have the computer modify your bonus based on the set of boxes on your screen, and enter this amount truthfully.

Next

After the NoCalc treatment, the subjects enter the Calc treatment and are shown the next instructions

Calculator Provided

You may have found that in doing the task you want to perform some calculation to the numbers. Accompanying all upcoming tasks, a calculator will be provided to you on your screen to help you.

Its up to you how you use the calculator. But **you need to submit the maximum amount that you are willing to pay using the calculator.** If you are chosen to be paid by the computer, you will be paid only according to the maximum amount you submit, not what you calculate.

How you submit your response in the calculator

For each task, we record your maximum amount willing to pay by taking the number in the *Result* column of the most recent line of your calculator as your response.

We now show you how to submit your response with the calculator using an example. **We deliberately choose an example unrelated to our task to avoid prejudicing you towards a particular way of valuing the boxes**.

Example: Imagine someone is asked to convert 32 kilograms to ounces, and submit their result in the calculator. They may do the following:

- First, they calculate $32 \times 2.205 = 70.56$ to convert 32 kilograms to 70.56 pounds (1 kilogram = 2.205 pounds)
- Second, they calculate $70.56 \times 16 = 1128.96$ to convert 70.56 pounds to 1128.96 ounces (1 pound = 16 ounces)
- Finally, they submit 1128.96 as their response.

We now show you how to achieve this in the calculator.

As a first step, you perform the steps as described above in the calculator until the most recent line in the *Result* column (Line 2 in this example) shows your intended response.

As a first step, you perform the steps as described above in the calculator until the most recent line in the *Result* column (Line 2 in this example) shows your intended response.

	Calculation	Result
1	32×2.205	70.56
2	70.56×16	1128.96
3		

You will be asked to double check whether the recorded response matches your intended response. If they do, click the "Submit" button, and then your response to this task will be recorded, and your will enter the next task. If they do not, you can modify the calculation steps until it leads to what you would like to submit.

	Calculation	Result
1	32×2.205	70.56
2	70.56×16	1128.96
3		

Ans	(	)	Del
7	8	9	/
4	5	6	×
1	2	3	−
0	.	Enter	+
Submit	You are about to submit **1128.96** as your response. Are you sure?		

For your task, you will similarly use the *Fill* and *Submit* buttons to submit the maximum amount that you are willing to pay to best advance your interests.

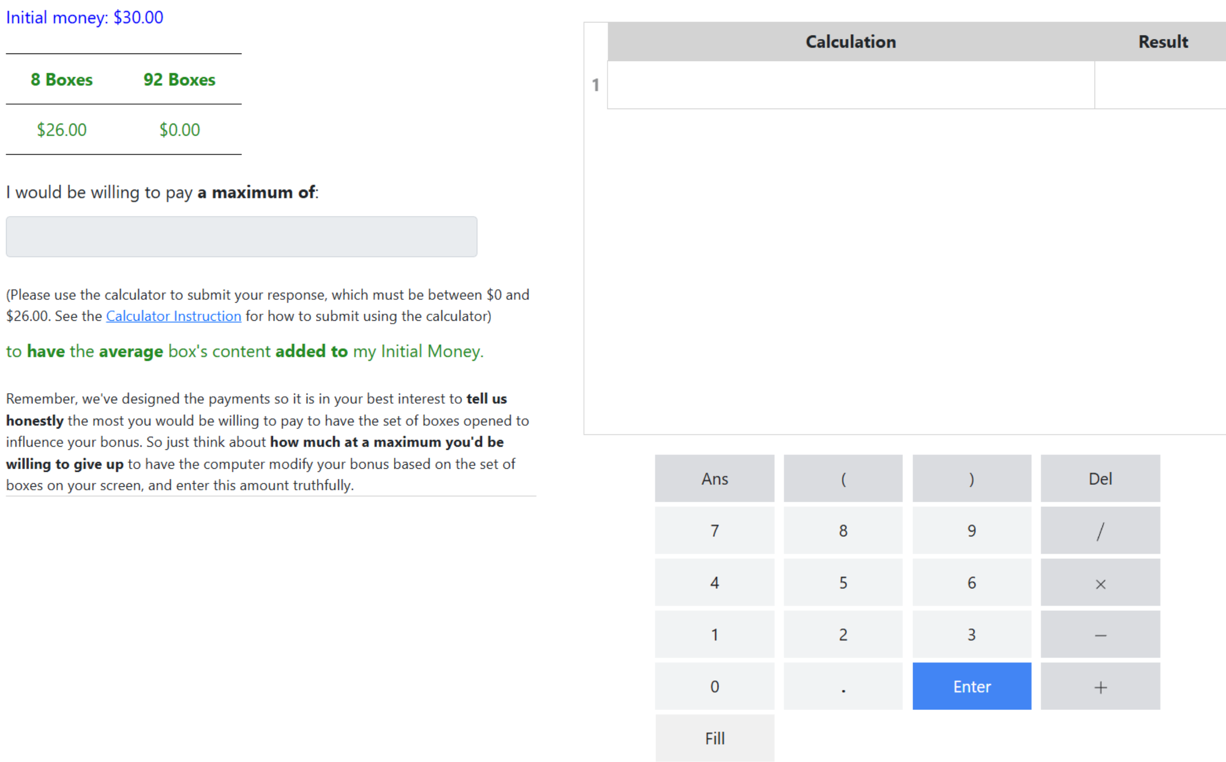
You may want to try the example above with the calculator provided below. **Also, you may want to try using the keyboard to type in the calculator.** For example, you can use `1` to type *1*, `*` to type ×, `Backspace` to type *Del*.

	Calculation	Result
1		

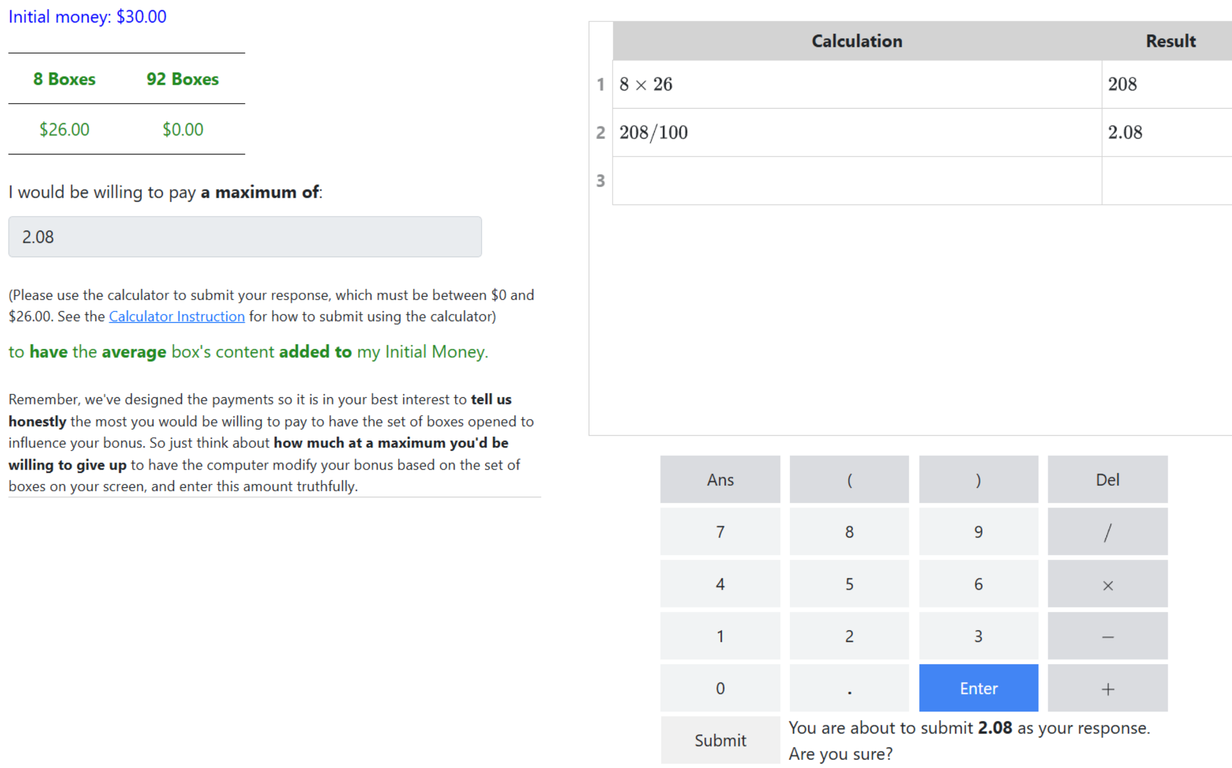
Ans	(	)	Del
7	8	9	/
4	5	6	×
1	2	3	−
0	.	Enter	+
Fill			

Next

Experimental interface in the Calc treatment, before the subject performs any calculation:



Experimental interface in the Calc treatment, after the subject performs some calculations and clicked *Fill*:



Reconciliation interface

Switcher Verbal Explanation

Choice task

EV Question